MASTER EN SCIENCES INFORMATIQUES

Analyse en vue d'une informatisation des données au niveau hospitalier

Poullet, Eddy

Award date:
1987

Awarding institution:
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix

ANALYSE EN VUE D'UNE INFORMATISATION

DES DONNEES AU NIVEAU HOSPITALIER

Promoteur : Jean FICHEFET

Mémoire présenté en vue de l'obtention du
grade de maître en informatique
par Eddy POULLET
1984-1987

INTRODUCTION

La Belgique possède une densité médicale peu commune. Son infrastructure contient un réseau tout aussi dense d'hôpitaux et de cliniques. On peut même ajouter que, dans certaines régions, il y a un excédent de "lits" par rapport aux besoins. De plus, les distances entre un point quelconque du territoire et les hôpitaux ou cliniques les plus proches sont habituellement courtes, comparées à des pays voisins. Si cela renforce la qualité du système de soins, le patient garde la possibilité de choisir l'institution de soins suivant des desiderata personnels mais aussi suivant les problèmes pour lesquels il y a recours.

La quasi-totalité des naissances se font en milieu hospitalier (dans le cas de notre pays); de même la majorité des décès s'y produisent. Dans le déroulement d'une vie, bon nombre de problèmes de santé nécessitent une hospitalisation. A tous ces évènements, de l'information à caractère médical est créée au sein de l'hôpital. Ces dernières années, au vu de la multiplicité des différents examens, la quantité d'information devient considérable. Tout cela amène aussi un surcroît de gestion administrative qui est étroitement liée à ces mêmes informations. Ce document se propose d'analyser et d'essayer de structurer tout ce flot de données pour les rendre compatibles avec un système d'information.

Le premier chapître explique la situation actuelle de ce problème. Un hôpital répond à une organisation particulière au sein de laquelle des informations de différents types circulent. Tout patient y possède un ou plusieurs dossiers médicaux qui seront archivés.

Le second chapitre, après les spécifications d'un hôpital-type, montre le développement d'un système d'information selon JSD ("Jackson System Development"). Cette méthode permet de concevoir de grands systèmes comme celui d'un hôpital entier. La partie du travail qui concerne le laboratoire d'analyses médicales est plus détaillée.

Le troisième chapître propose d'évaluer la méthode. Une brève description de la programmation "Object-oriented" est faite. Elle suggère d'utiliser ce type de programmation à des systèmes construits sous JSD.

Une réflexion de l'apport de JSD dans les techniques de développement clôture ce travail.

CHAPITRE 1 : ASPECT ACTUEL DU PROBLEME

1. ORGANISATION MEDICALE D'UN HOPITAL

Si les premiers médecins effectuaient à la fois la médecine (non chirurgicale), la chirurgie et les accouchements, l'évolution scientifique a, au début de ce siècle, fait apparaître des spécialités qui nécessitent une formation et une expérience bien définie. L'hôpital qui au départ se décomposait en de grandes salles avec de nombreux malades alités pour diverses raisons, a subi parallèlement à l'amélioration de l'hygiène et de la technique des multiples subdivisions. Actuellement, il est obligé de satisfaire à une organisation stricte. Les impératifs de rapidité et de sécurité (ne perdre ni un malade, ni un document, ni un échantillon à analyser) imposent à celle-ci d'être bien adaptée.

1.1. Les services

En ce qui concerne la partie médicale, tout hôpital ou clinique se divise en deux types de service.

1.1.1. Les services de soins

Ils accueillent les patients pour un ou plusieurs jours et leur prodiguent des soins que réclame leur état.
Deux grandes spécialités se détachent des autres par le volume de patients qu'elles prennent en charge : la médecine interne et la chirurgie.

Deux schémas d'organisation se dégagent principalement :

Premier schéma

Les spécialités de chirurgie et de médecine interne sont disjointes et deviennent deux grands départements séparés avec chacun plusieurs subdivisions.
Citons à titre d'exemple, en médecine interne :
- la gastro-entérologie
- la cardiologie
- l'endocrinologie
- la gériâtrie
- la néphrologie
- l'hématologie, etc...

et en chirurgie :
- chirurgie cardiaque
- chirurgie thoracique, abdominale (digestive)
- orthopédie, etc...

Chacune de ces subdivisions correspond à un service. Les autres spécialités reçoivent moins de patients et de ce fait, ont des petits départements ou même se regroupent en département par spécialité complémentaire. En règle générale, ils sont assimilés à des services.
Citons :
- la dermatologie
- la pédiatrie
- la neurologie
- la gynécologie
- l'obstétrique
- la psychiatrie
- l'ophtalmologie
- l'oto-rhino-laryngologie
- la cancérologie...

La pédiatrie cependant, équivaut aussi dans un hôpital de taille moyenne ou plus à un département et ceci devient compréhensible quand on réalise que cette tranche d'âge demande un "nursing" particulier. Elle se divise en :
- néonatalogie (les nouveaux-nés et prématurés ont besoin de couveuses et de soins différents)
- la neuropédiatrie, la chirurgie pédiatrique, la cardiologie pédiatrique se regroupent ensemble (les enfants ont des habitudes différentes, des nourritures différentes, etc...)

Il faut savoir que le nombre restreint d'enfants hospitalisés ramène ce département au rang d'un service quand on le compare avec la médecine interne ou la chirurgie.

Deuxième schéma

La distinction entre médecine interne et chirurgie n'existe plus. L'apparition de services médico-chirurgicaux provoque une division différente. Le service de gastro-entérologie contient à la fois des patients adressés à des internistes que des patients confiés aux chirurgiens (chirurgie digestive). Un parallélisme n'est pas toujours possible mais plusieurs services s'apparient bien. Les autres services ou départements discutés dans le premier schéma ne change pas d'organisation. Précis ons que toutes les variantes sont possibles. Certaines organisations fusionnent le service de gynécologie et le service d'obstétrique en un département de gynéco-obstétrique par exemple.

Quelques services de soins ne correspondent à aucune spécialité particulière et prennent en charge des malades pendant un temps assez bref habituellement. Cependant, les soins de "nursing" demandent une qualification et/ou du matériel spécialisé dans leur domaine respectif.
Ce sont :
- Service de soins intensifs (ou réanimation)
- Service d'urgence
- Service de soins stériles
- Service de grands brûlés

Le lecteur peut remarquer que le service de néonatalogie correspond bien aussi à cette définition où du matériel spécialisé est employé. Le pédiatre responsable (néonatalogue) est un spécialiste et il semble plus conforme de considérer ce service comme un service de soins. D'ailleurs, il existe dans certains hôpitaux une réanimation pédiatrique (où existent quelques couveuses) dans les soins intensifs. Dans ce dernier cas, une autre question surgit : la réanimation pédiatrique dépend-t-elle du département de pédiatrie ou des soins intensifs? Géographiquement et pour des besoins de logistique, elle est associée au service de soins intensifs mais des pédiatres la dirigent médicalement.

CARDIOLOGIE	CHIRURGIE CARDIAQUE
GASTRO-ENTEROLOGIE	CHIRURGIE DIGESTIVE
PNEUMOLOGIE	CHIRURGIE THORACIQUE
.....	.....
Département de Médecine Interne	Département de Chirurgie

Fig. 1 : Cas du premier schéma

CARDIOLOGIE	-	CHIRURGIE CARDIAQUE	Service de Cardiologie
GASTRO-ENTEROLOGIE	-	CHIRURGIE DIGESTIVE	Service de Gastro-entérologie
PNEUMOLOGIE	-	CHIRURGIE THORACIQUE	Service de Pneumologie
.....	-	.....	

Fig. 2 : Cas du second schéma

L'unité de base est l'unité de soins. Elle identifie un lieu et est défini géographiquement par construction. Elle contient un nombre déterminé de chambres à 1, 2 ou plusieurs lits.

1.1.2. Les services d'investigations techniques ou services techniques

Ils seront dénommés pour plus de facilités, les services d'examens techniques dans le reste de ce travail. Ils comprennent principalement les services de
- radiologie
- biologie clinique
- scintigraphie
- thermographie
- anatomo-pathologie
- ...

La radiologie et la biologie clinique ont des multiples subdivisions.

Pour la radiologie :
- scanner
- échographie
- radiologie digestive
- radiologie rénale
- radiologie osseuse
- ....

Pour la biologie clinique :
- hématologie
- microbiologie
- tests nucléaires (RIA)
- immunologie
- hormonologie
- ....

Chaque hôpital aura des services d'examens techniques qui se subdiviseront plus ou moins différemment. D'autre part, il existe de petites unités d'examen technique, qui dépendent d'une spécialité de façon assez nette (donc, d'un service de soins). Cependant, la quantité de patients en provenance de divers services a motivé le détachement en unité d'examens techniques séparée du service de soins.

- Electrocardiogramme	CARDIOLOGIE
- Epreuves respiratoires	PNEUMOLOGIE
- Electroencéphalogrammes	NEUROLOGIE
- Endoscopie digestive	GASTRO-ENTEROLOGIE
-	.....

Toute description d'un hôpital ne saurait être complète si on ne citait pas :

1. LES CONSULTATIONS qui ne sont pas un service proprement dit mais un endroit où tous les services sont représentés. Les patients qui doivent se faire opérer ou qui ont déjà été opérés, consultent les médecins spécialistes. Le médecin-traitant les envoie aussi pour avoir l'avis d'un spécialiste. Bien souvent, si les malades hospitalisés peuvent se déplacer, ils se rendent en consultation chez tel spécialiste quand cela est nécessaire, au niveau prévu pour les consultations. Les rapports entre le service de Médecine interne et la consultation de Médecine interne paraissent moins étroits qu'ils ne sont. Les mêmes médecins travailleront dans les deux "services". Signalons que les unités techniques sont souvent rattachées à la consultation correspondante. Ils se sont rapprochés de la consultation notamment pour des raisons de secrétariat.

2. LE SERVICE D'ANESTHESIE. N'ayant pas de lit à sa disposition, il ne peut être considéré comme un service de soins. Tout le monde comprendra que son utilité dans la structure ne peut être négligée. Ce service est situé à proximité du bloc opératoire.

3. LE SERVICE DE MEDECINE PHYSIQUE qui fait un travail immense, dont on oublie assez souvent l'apport sur le plan médical. En effet, les kinésithérapeutes sous la direction d'un spécialiste en médecine physique, prodiguent les soins prescrits dans toute l'institution hospitalière. Quand les patients sont incapables de se déplacer, ils sont traités dans leur service de soins. C'est un service qui ne possède pas de lit ni d'unité de soins. Remarquons quand même que les services de traumatologie et d'orthopédie fourniront la majeure partie de leur activité.

4. LE SERVICE DE RADIOTHERAPIE n'existe que dans certains hôpitaux et sert à effectuer le traitement que les spécialistes en cancérologie ont prescrit en utilisant des sources radioactives naturelles ou artificielles.

1.2. Informations médicales

1.2.1. Les données signalétiques

Nom, prénom, date de naissance, numéro de dossier (identifiant).

1.2.2. La demande d'examen

Chaque examen est précédé d'une demande justifiant le type d'examen et aussi la facturation. Une note de deux ou trois lignes précise le but de l'examen chez un patient. Ces renseignements facilitent grandement la tâche de l'investigateur. Les examens radiologiques profitent certainement de cette information jointe à la demande alors que la biologie clinique s'en passe le plus souvent. Un résultat est normal pour une biologie s'il est compris entre deux bornes. Certains états modifient quand même les valeurs de références.
A titre d'exemple, une patiente enceinte, un malade utilisant des médicaments particuliers entraîne pour certaines analyses des modifications.

1.2.3. Le protocole

L'évolution et la découverte scientifique contribue à mettre à la disposition du corps médical des instruments d'analyse de plus en plus fins et précis. De nouvelles techniques apparaissent. La nécessité s'est fait sentir très tôt de spécialiser des médecins dans ces divers domaines. Le protocole représente la réponse du médecin, responsable de l'examen pratiqué au médecin demandeur, via la demande d'examen. L'amortissement de l'appareil, le support de l'information, la prise des données ne sont d'ailleurs remboursés par l'organisme assureur que si les examens ont été protocolés. Il faut savoir, que, très souvent, c'est un personnel non médecin (technicien) qui enregistre les données. L'analyse d'un protocole sera présentée dans le second chapître. La distinction entre protocole de type texte et de type chiffre est évidente par définition. De plus, les protocoles de type chiffre s'accompagnent rarement de commentaires, tout comme leur demande. Le biologiste engage sa responsabilité sur la validité des résultats obtenus à partir des échantillons reçus.

1.2.4. La demande d'avis médical

La médecine avec les différentes spécialités s'est sectorisée.Les matières que chaque spécialiste connaît constituent un monde à part entière. Quand un malade se présente dans un service, une demande d'avis d'un confrère spécialiste est envoyée. Par exemple, dans le cas d'un patient hospitalisé pour un problème cardiaque, s'il présente une pathologie cutanée spéciale, le chef de service ou l'assistant demandera un avis au dermatologue.

1.2.5. La réponse d'avis médical

C'est l'équivalent du protocole accompagnant les données d'un examen technique. Le médecin consulté retourne au demandeur, un diagnostic et une proposition de traitement.

1.2.6. Le support d'examen

Toute une série d'examens techniques nécessite l'utilisation d'un support image : la radiographie, la scintigraphie, la thermographie, l'échographie, le scanner, les clichés par résonnance magnétique nucléaire, ECG, EEG, ... L'interprétation de ce support se fait par le médecin protocoleur. Il est intéressant de les avoir pour visualiser et comparer à d'autres clichés du même malade et de la même région. Certains examens utilisent des supports images-couleur : la scintigraphie, l'endoscopie et parfois la thermographie.Les documents de l'électrocardiographie et de l'encéphalographie s'apparentent plus à un tracé qu'à une image mais leur valeur en tant que support de données reste identique par rapport aux radios et assimilés.

1.2.7. Le protocole opératoire

Toute sanction chirurgicale chez un patient se réalise par une technique que le chirurgien emploie dans ce cas et pas dans un autre. De plus, des petits détails émaillent cet acte. Chaque chirurgien décrit au dictaphone l'opération qu'il a faite en précisant le plan d'abord de l'organe ou région, les fils utilisés, etc.. Ce document contient la description de l'opération que la secrétaire tape ensuite.

1.2.8. Les antécédents

Les antécédents précisent, comme leur nom l'indique, le passé médical qu'a présenté le patient lui-même mais aussi des membres de sa famille. Ils peuvent être familiaux, personnels, (médicaux), chirurgicaux, psychiques.

1.2.9. L'anamnèse ou données subjectives

Le médecin se fait l'interprète des plaintes du malade et les traduit sous forme de termes médicaux. D'autres questions que le practicien demande, l'aident à circonscrire le (les) problème(s) de santé. Ces données sont subjectives parce qu'elles mettent en cause la sensibilité d'un chacun. De plus, le médecin est obligé de croire ce que le malade lui révèle. On ne sait pas vérifier si une personne a réellement mal ou non.

1.2.10. Les données cliniques objectives

En d'autres mots, il s'agit de l'examen clinique. Il varie selon le médecin consulté, selon sa spécialité. On comprendrait mal qu'un cardiologue n'ausculterait pas un patient tandis qu'un dermatologue qui le fait toujours étonnerait (si pas plus) sa clientèle. Voici quelques éléments de l'examen clinique :

- inspection
- palpation
- auscultation
- percussion
- tension artérielle et pouls
- reflexe ostéo-tendineux (ROT)

Tous ces renseignements sont consignés et seront comparés lors d'un examen ultérieur.

1.2.11. Le diagnostic

L'ensemble des données que le médecin possède lui donne à un certain moment, un diagnostic répondant aux plaintes du patient, quand cela est possible.

1.2.12. Le traitement

Le patient reçoit un traitement qu'il devra suivre avec une posologie donnée pendant une certaine durée. Il est censé solutionner les ennuis présentés.

1.2.13. Lettre au médecin-traitant

Cet élément est capital. Elle reprend une synthèse de toute une hospitalisation : les plaintes à l'entrée, les différents examens cliniques et techniques, les conclusions diagnostiques et une solution thérapeutique adaptée. Si elle joue un rôle déontologique en assurant la continuité des soins, pratiquement c'est le seul document que le médecin-traitant recevra sur les différents examens que son patient a subis.

1.3. Le dossier médical

Le dossier médical est un ensemble structuré d'informations médicales. Au niveau hospitalier, il est le témoin de tous les évènements médicaux que le patient a vécus ou vit. Comment se présente-t-il à l'heure actuelle? A la suite des données signalétiques, s'aligneront les antécédents familiaux puis ceux du patient lui-même. Tous les éléments dominants de son passé et présent médical seront notés; ou bien, si le dossier existe déjà, les problèmes médicaux qui ont surgi depuis la dernière hospitalisation ou consultation. Une anamnèse précise la raison pour laquelle la personne s'est présentée. Le médecin transcrit les plaintes les plus significatives. A cela succède l'examen clinique qui révèle quelques anomalies ou non qui sont scrupuleusement répertoriées. Le médecin,s'il l'estime nécessaire, a rédigé quelques demandes d'examens complémentaires et d'avis de confrère, après avoir consulté les protocoles des examens et des avis qu'il avait demandé la fois précédente. Les éléments intéressants des protocoles et réponses d'avis de même que les nouveaux examens et avis demandés sont retranscrits dans le dossier. Le diagnostic et le traitement correspondant sont consignés. Les protocoles et réponses d'avis s'insèrent dans la farde du dossier et, en fait, ne sont repris sur les feuilles de travail que les faits relevés comme inhabituels. Dans le cas d'une hospitalisation, la structure du dossier médical reste identique. Chaque journée d'hospitalisation ou chaque intervention du médecin par journée (au moins une) équivaut à une nouvelle ouverture du dossier. Les différents points seulement ont moins d'éléments que le premier jour. Les antécédents du malade ne changent pas au cours de l'hospitalisation et les annotations d'une journée sont souvent plus courtes que celles d'une consultation. Une lettre au médecin-traitant conclut l'hospitalisation mais aussi bien souvent une consultation surtout dans les services universitaires.

En résumé, le dossier médical contient :

- une partie signalétique. Dans certains cas, la farde elle-même est référencée; dans d'autres, chaque patient est identifié par un "badge" que le personnel emploie pour marquer chaque feuille du dossier. Tout document égaré sera donc restitué au service concerné.
- une ou plusieurs feuilles de travail
- une ou plusieurs lettres au médecin-traitant
- les protocoles des examens techniques
- les réponses des avis demandés

Les feuilles de travail contiennent un ensemble de données répétitives :
- date, heure
- antécédents
- anamnèse
- données objectives (ex. clinique)
- réponse d'avis médicaux
- résultat des protocoles d'examens demandés
- nouvelles demandes d'examens
- nouvelles demandes d'avis
- évaluation diagnostic
- traitement

Ce groupe de données se répète au fil des consultations et des journées d'hospitalisation. Les cas aigüs requèreront plusieurs fois par jour ce groupe de données.

La décision du conseil médical et l'organisation de l'hôpital sont parfois à l'origine de la multiplicité des dossiers médicaux. Dans ce cas, chaque service possède un dossier médical par patient. Ce qui signifie qu'un patient peut avoir, pour une même pathologie, ayant nécessité un transfert de service, deux dossiers médicaux. Dans ce type d'organisation, il est fréquent qu'un patient a dans un service de médecine interne, un dossier pour un infarctus et dans un service de chirurgie, un dossier pour une fracture du col du fémur. Danq chacun des dossiers, on retrouvera un avis de service où il n'est pas hospitalisé : quand il s'est fait opérer pour son col du fémur, un avis cardiologique a été demandé; par contre, quand il est alité en médecine interne pour son infarctus, un avis orthopédique est demandé pour évaluer l'état de sa prothèse (relatif à son fémur). De plus en plus d'hôpitaux optent pour un dossier unique avec plusieurs fardes, une par département et l'accès à tout le dossier est possible.

1.4. Archivage

Le destin de tout dossier médical, tôt ou tard, est de se retrouver dans les archives. Les dossiers médicaux après hospitalisation ou après consultation, quand la lettre au médecin-traitant a été dictée, tapée par le secrétariat et corrigée par le médecin, sont dirigés vers le service d'archivage. Le volume des dossiers médicaux devient rapidement énorme. Il faut savoir qu'un hôpital de 900 lits (qui possède aussi un service de consultation pour patient ambulant) produit environ 20.000 nouveaux dossiers médicaux par an. Pour un hôpital de cette taille, les archives contiennent près d'un demi million de dossiers médicaux. Les examens techniques produisent des protocoles et des supports images. Le service de radiologie effectue cinq cent examens par jour. Chacun de ceux-ci nécessite jusqu'à 30 clichés. Les formats de ces derniers varient beaucoup. Si en général, les services techniques ont leur propre archivage pour les protocoles, les radiographies elles-même sont confiées à l'archivage central. Il faut faire remarquer que les radios montent dans les services, accompagnées des protocoles puis redescendent aux archives. Seulement, bien souvent, des clichés sont égarés ou subtilisés quand ils sont acheminés vers les archives.

1.5. Redondance

Les informations redondantes existent à plusieurs niveaux. La lettre au médecin-traitant est une information redondante. Tous les protocoles existent en double exemplaire dont un est conservé dans les archives du service technique. Parfois même, plusieurs médecins d'un hôpital reçoivent un exemplaire. Si les dossiers ne sont pas uniques, alors une troisième source, la plus grande, de redondance se crée.

2. ORGANISATION NON MEDICALE D'UN HOPITAL

2.1. Les services

Dans la partie non médicale, l'hôpital compte plusieurs services qui ont une relation directe ou indirecte avec la gestion. Ne sont impliqués dans ce projet que les services ayant un rapport plus privilégié avec le côté médical. Les services s'occupant de l'entretien, de l'hôtellerie, de la bibliothèque ne sont pas repris.

2.1.1. Le service administratif

Il se divise en plusieurs départements notamment l'accueil, la comptabilité, la facturation, etc...Bien que le service administratif gère toute l'administration de l'hôpital, seuls les départements cités seront concernés par ce travail.

2.1.2. La pharmacie

Elle est située dans la partie non médicale pour la bonne et simple raison que la part administrative est bien plus grande que la part médicale. On peut considérer que la pharmacie est une gestion de stock un peu particulière.

L'unité de commande de la pharmacie aux services (de soins ou techniques) n'est pas la même pour l'unité de facturation de l'hôpital au patient.
Exemple : Le service de soins reçoit de la pharmacie 1000 comprimés (un flacon) d'un médicament. Le patient reçoit sur sa facture X comprimés qu'il a reçus pendant son séjour et qui seront remboursés par son organisme assureur

2.2. Informations non médicales

2.2.1. Les données signalétiques

En plus des renseignements identifiant le patient, elles contiennent toutes les informations relatives à son organisme assureur.Tous les éléments permettant de tarifier et de facturer les prestations dont il a bénéficié à la fédération dont il est membre ou à une assurance indépendante.

2.2.2. Les prestations

Toutes les prestations effectuées ont une partie de renseignements administratifs (effectués en urgence, ou non, prestations sur un patient consultant ou hospitalisé,...).

2.2.3. L'hospitalisation

Le nombre de jours et la catégorie de chambre occupée, catégorie de la chambre demandée, le nombre de jours de soins sous la responsabilité de tel médecin doivent être précisés.

2.2.4. Les médicaments

Tout médicament qui n'est pas compris dans le prix de la chambre et qui est fourni par le service doit être tarifié.

2.3. Le dossier administratif

Il est en fait la somme des différentes factures détaillées des patients. Si le patient ou l'organisme assureur ne paient, et ceci pour diverses raisons, un dossier est ouvert par le service contentieux.

2.4. Archivage

Une fois tarifiées et facturées, toutes les données administratives sont archivées comme le veut la Loi. Il n'y a bien sur que les cas de conflits ou d'enquête administrative qui nécessiteront la remise dans le réel de ces informations.

CHAPITRE II : DEVELOPPEMENT D'UN LOGICIEL SOUS JSD

1. INTRODUCTION

Les principaux éléments d'analyse fonctionnelle d'un hôpital type sont repris dans un premier temps. Certaines options décrites dans le chapître précédent sont choisies pour pouvoir servir de spécification de développement d'un système d'information. Ce projet se propose de construire un logiciel pour la gestion d'un hôpital (données médicales et non médicales). L'orientation de ce travail est plus orientée vers la gestion d'un laboratoire d'analyses médicales mais en prévoyant dès le début l'intégration des autres points progressivement.

La méthode JSD est utilisée pour rencontrer les spécifications précisées. Différentes remarques émaillent les étapes du développement afin de mettre en évidence les particularités de la démarche.

2. SPECIFICATIONS DU PROJET

L'hôpital comprend un certain nombre de lits d'hospitalisation partagés entre plusieurs unités soit médicales, soit médico-chirurgicales. Chaque unité contient un nombre déterminé de chambres de différents types (particulière, à deux lits, à quatre lits, ...) La configuration des chambres et des lits est fixe.
L'hôpital comprend aussi d'autres services :
- les services d'examens techniques
- les consultations
- les services administratifs
- la pharmacie
- le bloc opératoire, soins intensifs et anesthésiques sont assimilés à des services de soins.

1. L'accueil du patient

Administrativement, tout patient qui a un contact avec un médecin doit passer par le service d'accueil. Ce contact peut être une consultation ou une hospitalisation. Le personnel de l'accueil s'occupe de recevoir les informations nécessaires du patient (nom, prénom, date de naissance, renseignements souhaités).
Un numéro de dossier est attribué et un "badge" est donné au patient.
Si le patient est déjà inscrit, les modifications à ses renseignements et un nouveau "badge" sont réalisés si nécessaire. Le service des urgences peut, quand l'accueil est fermé (week-end, nuit) effectuer les opérations minimales pour enregistrer un nouveau patient (les autres renseignements sont complètés plus tard).
Pour une hospitalisation, suivant la catégorie de chambre demandée et suivant les renseignements mutuelle fournis, un acompte est demandé au patient pour les frais non couverts.

Une admission n'est refusée que dans le cas où aucune chambre n'est libre dans le service demandé ou dans les services ayant la même vocation.

2. Le séjour du patient hospitalisé

ENTREE et SORTIE

Une fois le patient admis, il est dirigé vers un service de soins. Si le patient entre par la voie du service d'urgences, il sera aussi après un certain laps de temps si l'hospitalisation s'impose, conduit dans un service de soins approprié. Le "badge" du patient (reprenant l'identification du patient) est joint au dossier et servira à marquer toutes les demandes d'examen ainsi que les feuilles du dossier médical.

La journée d'hospitalisation commence après 14 heures et finit le lendemain avant 14 heures. Toute journée commencée est à considérer comme complète.
Les sorties provisoires ne sont pas admises. La sortie du patient libère le lit qu'il occupait. Un décès est un type particulier.
Pendant son hospitalisation, le patient peut être amené à suivre un régime prescrit par le médecin (ou demandé par l'infirmière responsable: par exemple un diabétique aura systématiquement un régime demandé par le responsable de la surveillance du patient).
Les renseignements concernant les médicaments utilisés pendant une hospitalisation sont encodés par période d'une semaine, à partir des feuilles de soins des infirmières.

TRANSFERT INTERNE (physique ou médical)

Le patient peut au cours d'une hospitalisation, changer de lit et cela pour diverses raisons. Il peut aussi recevoir une chambre de catégorie inférieure à celle qu'il demande et puis, par la suite, l'obtenir.De même, l'état de santé d'un patient peut justifier l'utilisation transitoire d'une chambre particulière alors que le patient ne l'a pas demandée.
Un patient hospitalisé est placé sous la responsabilité d'un médecin chef de service et de son équipe médicale (assitant, adjoint, stagiaire, ...).Il se peut que pour certaines raisons, il y ait un changement du responsable médical (appelé ici transfert médical). Dans le cas d'unité médico-chirurgicale, le malade change de médecin responsable sans changer de lit physiquement (problème médical qui se transforme en un problème chirurgical). Le transfert physique avec ou sans transfert médical est une situation tout aussi légitime.

LES PRESTATIONS

Elles sont présentées pour un patient par un médecin prescripteur - en général un membre de l'équipe médicale, plus rarement un médecin de garde. Un médecin prestataire les réalise. Dans certaines prestations, un rendez-vous doit être pris par le service de soins, vu les disponibilités matérielles de machines ou personnes. C'est le cas de la plupart des examens techniques. Les demandes d'avis qui sont aussi des prestations sont faites dans le service de soins du patient s'il ne peut se déplacer sinon celui-ci se rend (accompagné ou seul) à la consultation.
D'autre part, il se peut que le patient ait demandé de profiter de sa présence à l'hôpital pour consulter un médecin (un ophtalmologue par exemple) indépendamment du problème médical qui concerne son hospitalisation. Au cas où le rendez-vous existe avant, son hospitalisation (non prévue) est aussi possible. Une prestation peut être annulée.

En ce qui concerne la biologie clinique, un(e) employé(e) du service de soins prépare la veille les documents qui accompagneront les échantillons de sang prélevés le matin à jeun (sur demande d'un membre de l'équipe médicale). Ceux-ci seront acheminés avec le bordereau de demande aux différents laboratoires.

FICHE MEDICALE :

Il est décidé de recourir à une fiche médicale pour tous les patients. Si le médecin ne regarde (principalement) que les quelques mots-diagnostic dans un protocole, il faut se rendre compte que quand un malade présente un problème, il n'est pas intéressant pour lui d'avoir tout le dossier complet de A à Z, avec un dossier reprenant le plus de renseignements médicaux possible. Aussi l'idée d'une fiche médicale reprenant certaines données du patient répond à ce problème. Elle peut se transmettre à une autre institution de soins quand la vie d'un malade est en jeu (coma, imprécision,...).
Elle comprend deux parties :
La première reprend l'historique médicale du patient avec les antécédents, les hospitalisations, les consultations.
La seconde partie explicite les problèmes du malade à la dernière mise à jour.
Le médecin décide lui-même de l'intérêt de la mise à jour qu'il veut faire pour le patient qu'il a sous sa responsabilité.

3. Le patient ambulant

Il vient consulter un médecin spécialiste ou subir un examen technique à la demande d'un médecin. La prescription d'un médecin n'est pas obligatoire pour une consultation. Par contre, la prescription pour un examen technique (RX, ...) est toujours indispensable. Parfois, une réservation est aussi indispensable. Au cours d'une consultation, un médecin peut demander un ou plusieurs examens techniques, une ou plusieurs demandes d'avis, avant de pouvoir émettre un diagnostic. La demande d'avis correspond à une consultation en fait.

N.B. Le patient ambulant qui vient pour des soins, doit passer par l'accueil où il reçoit une feuille qui sera associée à son dossier. Après la consultation, son code et le code des actes techniques associés à celle-ci, sont consignés sur le document qui servira à la tarification puis à la facturation.

4. Les services d'examens techniques

Les services d'examens techniques sont tous les services qui produisent des protocoles suite à une demande d'un médecin. Dans certains cas, le médecin prestataire est le même que le médecin prescripteur (par exemple : l'ECG).
Une fois l'examen réalisé, les données recueillies sur un support sont analysées et protocolées. Des services tels que la radiologie, la scintigraphie, ... doivent s'organiser et de ce fait, des réservations devront être prises. D'autres examens nécessitent aussi un produit médical qui sera facturé au patient.

Les supports principalement des clichés RX ou des photos sont remis au médecin prescripteur. Dans le cas du patient hospitalisé, ils seront stockés soit avec son dossier, soit dans les archives, par type d'examen, après avoir été examinés par l'équipe médicale du service de soins.

5. Le service de soins

L'hôpital-type qui sert d'hypothèse de travail, comprend plusieurs services de soins. Certains de ceux-ci ont plusieurs unités de soins. D'autres, par contre, se partagent une même unité de soins. La répartition des lits dans ce dernier cas n'est pas fixe et varie suivant les nécessités du moment. En principe, un patient se trouve toujours dans l'unité de soins ad hoc mais il se peut que parfois, pour une courte période de temps, le lit attribué soit dans une unité de soins qui ne correspond pas au service de soins auquel il est rattaché.

6. Le service de biologie clinique

Les échantillons sont acheminés au laboratoire avec une demande de laboratoire qui comporte les informations signalétiques et les analyses à effectuer. Certaines peuvent être demandées en urgence. Au laboratoire, les échantillons et les demandes sont étiquetées avec un numéro unique (identifiant). Toutes les analyses pour chaque section du laboratoire, sont triées et l'échantillon de sang réparti suivant les besoins des laborantins.

Après que les analyses soient réalisées, les résultats sont retranscrits sur les protocoles avec les valeurs de référence et parfois un commentaire. Les résultats sont validés par le biologiste avant d'être envoyés à l'équipe médicale ou au médecin demandeur. Les analyses urgentes sont effectuées et validées le plus rapidement possible avant d'être envoyées au médecin demandeur.

7. La tarification et la facturation

Une fois par mois, toutes les prestations qu'un malade a reçues par service (de soins ou autre) sont regroupées. Le service administratif se charge de les tarifier et de les facturer aux différents organismes assureurs.

3. DEVELOPPEMENT DU SYSTEME

La méthode JSD s'applique particulièrement à ce problème. Toutes les informations à l'intérieur d'un hôpital circulent dans différentes directions. C'est une "réalité dynamique" qui se prête bien à être développée selon les concepts de cette démarche. Un résumé de la méthode avec quelques points caractéristiques est joint à ce travail (Annexe 1).

Les notes qui suivent reprennent le résultat final du développement. Bien souvent, quelques rétroparcours ont été nécessaires pour y parvenir. Autant que cela est possible, des commentaires expliquent ce qui de prime abord peut paraître fautif ou inadéquat. Cette méthode s'appuie considérablement sur l'outil graphique. Celui-ci en permet l'exploitation maximale et facilite l'appréhension des processus surtout quand le nombre des entités augmente.
Par contre, la présentation d'un travail ne s'accorde pas toujours facilement à l'adjonction de multiples schémas (taille, situation par rapport au texte associé, replication de la même structure à quelques détails près, par exemple).
Il semblait plus logique d'opter pour un document séparé ne contenant que les chémas.
Dans les listes des entités et les descriptions des actions, l'ordre alphabétique a été choisi arbitrairement par mesure de commodité.

PREMIERE PARTIE : CONSTRUCTION DU MODELE

3.1. ENTITY ACTION STEP

ENTITY AND ACTION LISTS

ADM (Administration) :
ATTRIBUER, DOS-FERMER, DOS-OUVRIR, ENREGISTRER, LIBERER, SIGN-MODIFIER, TRANSFERER

ANALDS (Analyse-descripteur) :
A-ENREGISTRER, A-MODIFIER

CELEM-DOS (Comptabilité-élement-dossier) :
DOS-COMPTABILISER, DOS-FERMER, DOS-OUVRIR, DOS-PRECOMPTABILISER

CHOSPITALISATION (Comptabilité-hospitalisation) :
ATTRIBUER, H-COMPTABILISER, H-PRECOMPTABILISER, LIBERER

DA-EX (Demande-Analyse-Examen) :
DA-ANNULER, DA-INTRODUIRE, DA-MODIFIER

DEM-A-PROV (Demande-Analyse-Provisoire) :
DA-PRO-ACCEPTER, DA-PRO-ANNULER, DA-PRO-INTRODUIRE, DA-PRO-REFUSER

DEM-L-PROV (Demande-Labo-Provisoire) :
DL-PRO-ACCEPTER, DL-PRO-ANNULER, DL-PRO-INTRODUIRE, DL-PRO-REFUSER

DL-EX (Demande-Labo-Examen) :
DL-INTRODUIRE, DL-COMPTABILISER, DL-INTRODUIRE, DL-MODIFIER, DL-SUPPRIMER

D-MEDICAMENT :
CONSOMMER, M-COMPTABILISER, M-MODIFIER

DP-EX (Demande-Protocole-Examen) :
DP-ANNULER, DP-COMPTABILISER, DP-INTRODUIRE, DP-MODIFIER, DP-VALIDER

ELEM-DOS (Elément-Dossier) :
ANT-INTRODUIRE, CLOTURER, CONCLUSION-MODIFIER, DOS-FERMER, DOS-OUVRIR, FM-MODIFIER, L-REDIGER, RELATER

FMEDICALE (Fiche médicale) :
ENREGISTRER, FM-METTRE-A-JOUR

HOSPITALISATION :
ATTRIBUER, LIBERER, TRANSFERER

JOBLIST :
JL-ETABLIR, JL-SUPPRIMER

LABO (Laboratoire) :
A-MODIFIER, DA-ANNULER, DA-INTRODUIRE, DA-MODIFIER, DL-INTRODUIRE, DL-MODIFIER, ENREGISTRER, MED-ENREGISTRER, MED-MODIFIER, URG-VALIDER

LIT :
ATTRIBUER, LIBERER

MEDECIN :
MED-ENREGISTRER, MED-MODIFIER

PATIENT :
ATTRIBUER, CONSOMMER, DL-INTRODUIRE, DOS-FERMER, DOS-OUVRIR, DP-INTRODUIRE, ENREGISTRER, LIBERER, SIGN-MODIFIER, TRANSFERER

SERV-SOINS :
ANT-INTRODUIRE, CONCL-MODIFIER, CLOTURER, CONSOMMER, DL-INTRODUIRE, DP-INTRODUIRE, FM-MODIFIER, M-MODIFIER, RELATER

SERV-TECH :
CONSOMMER, DP-INTRODUIRE, DP-MODIFIER, DP-SUPPRIMER, DP-VALIDER, M-MODIFIER

URG-ANAL (Analyse urgente) :
URG-ANNULER, URG-INTRODUIRE, URG-MODIFIER, URG-TRANSMETTRE, URG-VALIDER

ACTION : DESCRIPTIONS

A-ENRE : Une analyse est enregistrée avec ses paramètres de laboratoire
action de ANALDS, LABO
attribut : Analyse-id
Analyse-libellé
Analyse-Mnemonique
Code prestation
Type de résultat
Valeurs de référence
Unité de mesure
Valeurs extrêmes
Urgence-Flag
Prestataire
Type d'analyse
Section Labo
Dimension
Taille Temps
Temps-Défaut
Séquence-Protocole
Libellé-Protocole

A-MDF : Une analyse avec ses paramètres est modifiée
action de ANALDS, LABO
attribut : idem que A-Enre mais sans l'identifiant

ANT-INTRO : Les antécédents du patient depuis son dernier contact médical sont décrits par le personnel du Serv-Soins (médicaux, secrétaire)
action de SERV-SOINS, ELEM-DOS
attribut :

ATTRIBUER : Un lit est attribué à un patient par le service administratif
action de ADM, PATIENT, LIT, HOSPITALISATION, CHOSPITALISATION
attribut : Patient-id
Dates d'attribution
Catégorie chambre demandée
Service du Pat
Acompte
variables dégénérées de LIT : lit-id
cat-chambre

CONCLUSION-MDF : Le service de soins modifie la conclusion d'un élément de dossier
Action de SERVS, ELEM-DOS
Attribut : Date
Texte

CONSOMMER : Un patient consomme des médicaments qui lui sont fournis par un service de soins ou un service technique
Action de PATIENT, SERVS/SERVT, D-MEDICAMENT
Attribut : Patient-id
Med prescripteur id
Date an-période
Médicament-id
Quantité

DA-ANNULER : Une demande d'analyse est annulée par le laboratoire
Action de LABO, DA-EX
Attribut : Date/heure
Encodeur-id

DA-INTRODUIRE : Une demande d'analyse est introduite par le laboratoire pour une demande de laboratoire
Action de LABO, DL-EX, DA-EX
Attribut de : Analyse-id
Date/heure
Encodeur-id
Flag-état (urgent, gratuit, ajouté par le biologiste,...)
Flag-résultat (provisoire, à vérifier, définitif,..)

DA-PRO-ACC : Une demande provisoire d'analyse est acceptée par le laboratoire
Action de FVERDEMPROV, DEM-A-PROV
Attribut : Date/heure
Encodeur-id

DA-PRO-ANN : Une demande provisoire d'analyse est refusée par le laboratoire
Action de FVERDEMPRO, DEM-A-PROV
Attribut : Date/heure
Encodeur-id
Motif

DA-PRO-INTRO : Une demande d'analyse provisoire est introduite par une demande de laboratoire provisoire par un service de soins
Action de DEM-A-PRO, DEM-L-PRO, SERVS
Attribut : Dem-Prov-id
Analyse-id
Date/heure

DA-PRO-REF : Une demande d'analyse provisoire est refusée par le laboratoire
Action de DEM-A-PRO, FVERDEMPRO

DA-MDF : Le laboratoire modifie la demande d'analyse
Action de LABO, DA-EX
Attribut : idem que DA-INTRO sauf analyse-id

DEM-PRO-VERIFIER : La vérification d'une demande provisoire de laboratoire est activée par le laboratoire
Action de LABO, FVERDEMPRO
Attribut : Dem-Prov-id

DL-INTRO : Une demande est introduite pour un patient par le laboratoire
Action de DL-EX, PATIENT, LABO
Attribut : Dem-id
Patient-id
Lit-id
Med-prescripteur
Status-Impression/date
Status-Protocole/date
Renseignement clinique

DL-COMPTABILISER : Comptabilisation des analyses d'une demande
Action de DL-EX
Attribut : Date/heure
No de comptabilisation

DL-MDF : Une demande de laboratoire est modifiée par le laboratoire
Action de LABO, DL-EX
Attribut : Patient-id
Lit-id
Med-prescripteur
Rens cliniques
Encodeur-id

DL-PRO-ACC : Une demande de laboratoire provisoire est acceptée
Action de FVERDEMPRO, DEM-L-PRO
Attribut : Date/heure d'acceptation
Encodeur-id

DL-PRO-ANN : Une demande de laboratoire est annulée par le service de soins qui l'a produite
Action de SERVS, F-VER DEM PRO, DEM-L-PRO
Attribut : Date/heure du refus
Encodeur/id

DL-PRO-INTRO : Une demande de laboratoire provisoire est introduite par le service de soins pour un patient
Action de DEM-L-PRO, SERVS, PATIENT
Attribut : Dem-Pro-id
Patient-id
Lit/service
Date demande/heure

DL-PRO-REF : Une demande de laboratoire provisoire est refusée
Action de FVERDEMPRO, DEM-L-PRO
Attribut : Date/heure de refus
Encod-id

DL-ST-MDF : Le status d'impression et le status de protocole sont modifiés par la fonction qui produit les protocoles
Action de FPROTO, DL-EX
Attribut : Status-Imp/date
Status-Protocole/date

DL-SUPPRIMER : Suppression d'une demande par le laboratoire
Action de LABO, DL-EX
Attribut : Date
Encod-id

DOS-COMPTABILISER : Comptabilisation d'un élément de dossier fermé
Action de CELEM-DOS, COMPTA
Attribut : Date/heure
Numéro de comptabilisation

DOS-FERMER : Fermeture d'un élément de dossier médical par l'administration
Action de PATIENT, CELEM-DOS, ELEM-DOS, ADM
Attribut : Date/heure

DOS-OUVRIR : Ouverture d'une élément de dossier médical par l'administration
Action de PATIENT, ELEM-DOS, ADM, CELEM-DOS
Attribut : Patient-id
Elem-dos-id
Date/heure
Medecin responsable
Type (hospitalisé ou consultation)
Référence-elem-dos

DOS-PRECOMPTA : Comptabilisation partielle d'un élément de dossier non encore fermé
Action de CELEM-DOS, FCOMPTA
Attribut : Date/heure
numéro de comptabilisation

DP-ANNULER : Annulation d'une demande d'examen par le service technique
Action de SERV-TECH, DP-EX
Attribut : Date
Encod-id

DP-COMPTABILISER : Comptabilisation de l'examen technique
Action de DP-EX, FCOMPTA
Attribut : Date/heure
numéro de comptabilisation

DP-INTRO : Une demande d'examen technique est introduite pour un patient
Action de DP-EX, PATIENT, SERVS
Attribut : Patient-id
Lit
Ex-Technique-id
Date de l'examen
Médecin prescripteur
Urgence-Flag
Référence-elem-dos

DP-MDF : Modification du protocole d'un examen technique par le médecin prescripteur (service technique)
Action de DP-EX, SERVT
Attribut : Date/heure
Medecin protocoleur
Texte

DP-VALIDER : Un examen technique qui est protocolé définitivement est validé par un médecin d'un service technique
Action de DP-EX, SERVT
Attribut : Date/heure
Medecin-valideur

ENRE : A chaque patient qui vient pour la première fois à l'hôpital, l'accueil (Adm) prend note des renseignements administratifs (mutuelle); une fiche médicale est créée
Action de ADM, PATIENT, FMEDICALE
Attribut : Nom-Prénom
Adresse
Date de naissance
Patient-id
Renseignements mutuelle
Date d'enregistrement

FM-MAJ : Une fiche médicale est mise à jour avec les modifications introduites pour celle-ci dans ELEM-DOS
Action de FMEDICALE, FMAJFM, ELEM-DOS
Attribut : Date
Texte

FM-MDF : Une fiche médicale pour les ELEM-DOS est modifiée par le service des soins
Action de SERVICE SOINS, ELEM-DOS
Attribut : Date
Texte

HISADM-CHERCHER : Activation de la recherche d'un historique administratif d'un patient
Action de LABO, FHISADM
Attribut : Patient-id

HISMED-CHERCHER : Activation de la recherche d'un historique médical d'un patient
Action de LABO, FHISMED
Attribut : Patient-id
Analyse-id (1 ou plusieurs)

H-PRECOMPTABILISER : Comptabilisation partielle d'une hospitalisation en cours
Action de CHOSPITALISATION, FCOMPTA
Attribut : Date/heure
numéro de comptabilisation

JOBLIST-DECLENCHER : Activation de la génération des joblists et de leur impression
Action de LABO, FJOBLIST
Attribut : numéro de joblist (1 ou plusieurs)
Type de joblist

JL-ETABLIR : Une joblist est introduite
Action de FJOBLIST, JOBLIST
Attribut : Tableau (pat, demande d'analyse)
Date
Numéro-id

JL-SUPPRIMER : Une joblist est supprimée
Action de FJOBLIST
Attribut : date

LIBERER : Un lit est libéré par un patient lors de sa sortie d'hospitalisation ou lors d'un transfert
Action de ADM, PATIENT, LIT, HOSPITALISATION
Attribut : id-Pat
Date

L-REDIGER : Le médecin (service de soins) rédige une lettre (au médecin traitant)
Action de SERVS, ELEM-DOS
Attribut : Elem-Dos-id
Texte

M-COMPTABILISER : Comptabilisation des médicaments
Action de D-MEDICAMENT, FCOMPTA
Attribut : Date/heure
Numéro de comptabilisation

M-MODIFIER : Modification des données relatives à la consommation d'un médicament
Action de D-MEDICAMENT, SERVT/SERVS
Attribut : Nom/Prénom
Adresse
Type (hospitalier ou externe)

MED-ENRE : Un médecin qui prescrit des demandes d'examens ou d'analyses est enregistré par la laboratoire ou le service technique
Action de MEDECIN, LABO/SERVT
Attribut : Nom/Prénom
Adresse
Med-id
Type (hospitalier ou externe)

MED-MDF : Modification des données relatives à un médecin par le laboratoire
Action de MEDECIN, LABO/SERVT
Attribut : Nom/prénom
Adresse
Type

PROTO-EDITER : Activation de la production et de l'impression des protocoles
Action de LABO, FPROTO
Attribut : Dem-id-début
Dem-id-fin
Date Début
Date fin

RELATER : Un médecin (service de soins) relate les différents éléments médicaux qu'il a recueillis auprès du patient
Action de ELEM-DOS, SERVS
Attribut :

SIGN-MDF : Les renseignements signalétiques du patient sont modifiés par l'administration (par exemple : accueil)
Action de ADM, PATIENT
Attribut : Nom-prénom
Adresse
Date de naissance
Renseignements mutuelle
Date d'enregistrement

TRANSFERER : Un patient est transféré d'un lit dans un autre lit
Action de ADM, PATIENT, HOSPITALISATION
Attribut : Lit-Origine
Lit-destination
Date
Catégorie demandée

TRANS-DECLENCHER : La transmission des résultats urgents validés est activée
Action de LABO, FTRANSMISSION
Attribut :

URG-ANNULER : Une analyse urgente est annulée
Action de URG-ANA, FURG
Attribut : Date/heure
Encod-id

URG-INTRO : Une analyse urgente est introduite
Action de URG ANA, FURG
Attribut : Anal-id
Demand-id
Date/heure Encodage
Encod-id

URG-MDF : Une analyse urgente est modifiée (Résultat)
Action de URG-ANA, FURG
Attribut : Résultat
Date/heure MDF
Encod-id

URG-TRANSMETTRE : Une analyse urgente est transmise
Action de URG-ANA, FTRANS
Attribut : Date/heure

URG-VALIDER : Le médecin du laboratoire valide les résultats de l'analyse demandée en urgence
Action de LABO, URG-ANA
Attribut : Date
Médecin valideur

Certaines actions qui sont des déclenchements des fonctions majeures comme FPROTO, FHISADM, FTRANS,... ont déjà été introduites dans le type d'entité qui en assure logiquement l'activation.
Certaines fonctions qui seront automatisées dans le développement sont déjà indiquées (comme étant titulaires d'action) : FCOMPTA, FMAJFM,...
Lors d'un rétroparcours, elles sont normalement ajoutées à cette étape.

2. ENTITY STRUCTURE STEP

Les acteurs principaux sont le patient (PATIENT), la demande d'examen technique (DP-EX), les médicaments (MEDICAMENT), la demande du laboratoire (DL-EX) qui contient une ou plusieurs analyses (DA-EX). Les demandes d'analyses urgentes (URG-ANA) nécessitent un traitement particulier.

L'hospitalisation (HOSPITALISATION) d'un patient lui associe un ou plusieurs lits (LIT). S'il est hospitalisé en ambulant, un dossier médical est tenu (ELEM-DOS) à chaque fois qu'il a un contact médical.

Les entités HOSPITALISATION, DP-EX, ELEM-DOS, DL-EX n'ont pas pu rester dans PATIENT parce que leurs actions sont concurrentes.
La sortie de l'entité CHOSPITALISATION à partir de HOSPITALISATION permet à celle-ci de n'avoir qu'une seule occurence par patient. Le cas n'est pas le même pour ELEM-DOS qui peut ne pas être cloturé alors qu'une autre occurence ELEM-DOS est déjà ouverte. De plus, il faut pouvoir aussi satisfaire la spécification qu'une personne hospitalisée (donc avoir une occurence d'ELEM-DOS ouverte) consulte un autre médecin (peut-être d'ailleurs à la demande du médecin responsable), ce qui automatiquement ouvre une nouvelle occurence d'ELEM-DOS.

Les types d'entité ADMINISTRATION, LABORATOIRE, SERV-SOINS, SERV TECHNIQUE, sont des entités qui ne servent qu'à transmettre des actions.
On remarque que beaucoup d'entités ont des actions qui seront produites par des fonctions (toutes les actions COMPTABILISER, PRECOMPTA) notamment.

Le type d'entité ELEM-DOS est la plus compliquée. Elle contient tous les renseignements médicaux que le médecin responsable peut trouver chez un patient en cours d'hospitalisation ou au cours d'une consultation. Un attribut précisera si l'ELEM-DOS correspond à une hospitalisation ou à une consultation (l'ouverture et la fermeture d'une occurence d'ELEM-DOS pour une consultation est faite par la secrétariat au retour de la feuille reçue à l'accueil). Il faut bien se rendre compte que les actions du modèle sont décrites dans l'ordre où elles se produisent dans le monde réel. Dans le cas présent, il est clair que la fermeture du dossier peut se faire avant l'action RELATER bien que les attributs de cette action soient relatifs à des évènements antérieurs.

Les schémas décrivent la structure des différents types d'entité. Les actions normalement "découvertes" à l'étape 4 y sont déjà incluses. Elles sont hachurées.

3.3. INITIAL MODEL STEP

Dans les pages qui suivent, tous les processus sont décrits en pseudo-langage. Certaines variables générales ont été rajoutées dans certains processus.

Il faut préciser qu'une entorse a été faite à la théorie. Une action doit être considérée comme atomique. Or l'action TRANSFERER a été, au niveau du type d'entité HOSPITALISATION, répartie entre deux actions (LIBERER puis ATTRIBUER), qui se retrouvent elle-mêmes dans l'entité HOSPITALISATION. En fait, on peut estimer que la signification des actions est différente au niveau des types d'entités HOSPITALISATION et LIT.

Il aurait fallu faire correspondre :

HOSPITALISATION	LIT
ENTRER	ATTRIBUER
TRANSFERER	LIBERER + ATTRIBUER
SORTIR	LIBERER

mais comme les actions ENTRER et SORTIR se superposent respectivement à ATTRIBUER et LIBERER, la substitution est acceptable (bien que non orthodoxe).

Rem : Les processus de level-1 sont décrits dans les pages suivantes en pseudo-langage.
Ceux de level-2 incluant les actions des fonctions sont décrits à la fin de 3.4.

```
ADM-1 seq
   read A
       ADM-1-BODY its
         ADM-ACTION sel (ENREGISTRER)
           ENREGISTRER, write ENREGISTRER to AP
           read A
         ADM-ACTION alt (ATTRIBUER)
           ATTRIBUER, write ATTRIBUER to AP
           read A
         ADM-ACTION alt (TRANSFERER)
           TRANSFERER, write TRANSFERT to AP
           read A
         ADM-ACTION alt (LIBERER)
           LIBERER, write LIBERER to AP
           read A
         ADM-ACTION alt (DOS-OUVRIR)
           DOS-OUVRIR, write DOS-OUVRIR to AP
           read A
         ADM-ACTION alt (DOS-FERMER)
           DOS-FERMER, write DOS-FERMER to AP
           read A
         ADM-ACTION alt (SIGN-MDF)
           SIGN-MDF, write SIGN-MDF to AP
           read A
         ADM-ACTION end
       ADM-1-BODY end
ADM-1 end
```

```
ANAL-DS-1 seq
        read L
        A-ENRE
        read L
        ANAL-ACTION itr
             A-MODIFIER
             read L
        ANAL-ACTION end
ANAL-DS-1 end
```

```
CELEM-DOS-1 seq
        read DC
        DOS-OUVRIR
        read DC
        DOS-FERMER
CELEM-DOS-1 end
```

```
CHOSPITALISATION-1 seq
        read HC
        ATTRIBUER;
        read HC
        LIBERER
CHOSPITALISATION-1 end
```

```
DA-EX-1 seq
        read LA
        DA-INTRODUIRE
        read LA
        DA-ACTION itr
          DA-MODIFIER
          read LA
        DA-ACTION end
        DA-ANNULER
DA-EX-1 end
```

```
DEM-A-PRO-1 seq
        read LAP
        DA-PRO-INTRO
        read SAP
        DEM-A-BODY seq
                DA-PRO-ANN
        DEM-A-BODY end
DEM-A-PRO-1 end
```

```
DEM-L-PRO-1 seq
        read PDLP
        DL-PRO-INTRO
        read SLP
        INTR-ANAL itr
                DA-PRO-INTRO; write DA-PRO-INTRO to LAP
        INTRO-ANAL end
DEM-L-PRO-1 end
```

```
DL-EX-1 seq
        read PDL
        DL-INTRODUIRE
        read LDL
        DL-BODY itr
                DL-ACTION sel (DA-INTRO)
                        DA-INTRODUIRE : write DA-INTRODUIRE to LA
                        read LDL
                DL-ACTION alt (DL-MDF)
                        DL-MODIFIER; write DL-MODIFIER to LA
                        read LDL
                DL-ACTION alt (DL-ST-MDF)
                        DL-STATUS-MODIFIER
                        read LDL
                DL-ACTION end
        DL-BODY end
        DL-TERMINAISON seq
                DL-SUPPRESSION
        DL-TERMINAISON end
DL-EX-1 seq
```

```
D-MEDICAMENT-1 seq
        read PM
        CONSOMMER;
        read SS&ST
        M-ACTION itr
                M-MDF;
                read SS&ST
        M-ACTION end
D-MEDICAMENT-1 end
```

Rem : SS&ST est un "rough merge"

```
DP-EX-1 seq
        read PDP
        DP INTRODUIRE
        read TDP
        DP-BODY sel (DP-ANNULER)
             DP-TERM seq
                 DP-ANNULER
             DP-TERM end
        DP-BODY alt (else)
             DP-ACTION seq
                  RESULTAT itr (not DP-VALIDER)
                    DP-MDF
                    read TDP
                  RESULTAT end
                    DP-VALIDER
             DP-ACTION end
      DP-BODY end
DP-EX-1 end
```

```
ELEM-DOS-1 seq
        read SD&PD
        DOS-OUVRIR; write DOS-OUVRIR to DC
        read SD&PD
        E-DOS-BODY itr (not DOS-FERMER)
             E-DOS-ACTION sel (ANT INTRODUIRE)
                  ANT-INTRODUIRE;
                  read SD&PD
             E-DOS-ACTION alt (RELATER)
                  RELATER;
                  read SD&PD
              E-DOS-ACTION alt (L-REDIGER)
                  L-REDIGER;
                  read SD&PD
              E-DOS-ACTION end
        E-DOS-BODY end
        DOS-FERMER, write DOS-FERMER to DC
        read SP&PD
        E-DOS-FIN itr (not CLOTURER)
              FIN-ACTION sel (ANT-INTRODUIRE)
                  ANT-INTRODUIRE,
                  read SD&PD
              FIN-Action alt (CONCL-MDF)
                  CONCL-MDF,
                  read SD&PD
              FIN-ACTION alt (FM-MDF)
                  FM-MDF;
                  read SD&PD
              FIN-ACTION alt (RELATER)
                  RELATER,
                  read SD&PD
              FIN-ACTION alt (L REDIGER)
                  L-REDIGER,
                  read SD&PD
              FIN-ACTION end
        E-DOS-FIN end
        CLOTURER
ELEM-DOS-1 end
```

Rem : SD&PD est un "rough merge"

```
FICHE-MED-1 seq
        read PF
        ENREGISTRER
        read DF
        MAJ-ACTION itr
                FM-MAJ;
                read DF
        MAJ-ACTION end
FICHE-MED-1 end



HOSP-1 seq
       nb-hosp = 0
       HOSP-SECTION itr
            HOSPITAL-BODY seq
                  read PH&TGM-DAY-HOSP
                  ATTRIBUER; write ATTRIBUER to HL
                             write ATTRIBUER to HC
                  nb-jour-hosp = 1
                  nb-hosp = nb hosp + 1
                  read PH&TGM-DAY-HOSP
                  HOSPITAL-ACTION itr
                        HOSPITAL-EVENT sel (not TGM-DAY-HOSP)
                              TRANSFERER; write LIBERER to HL
                                          write LIBERER to HC
                                          write ATTRIBUER to HL
                                          write ATTRIBUER to HC
                              read PH&TGM-DAY-HOSP
                        HOSPITAL-EVENT alt (TGM-DAY-HOSP)
                              TGM-DAY-HOSP
                              nb-jour-hosp = nb-jour-hosp + 1
                              read PH&TGM-DAY-HOSP
                        HOSPITAL-EVENT end
                   HOSPITAL-ACTION end
                   LIBERER; write LIBERER to HL
                            write LIBERER to HC
                   read PH&TGM-DAY-HOSP
            HOSPITAL-BODY end
       HOSP-SECTION end
HOSP-1 end


Rem : PH&TGM-DAY-HOSP est un "rough merge"
```

```
JOBLIST-1 seq
     read FJL
     JL-ETABLIR
JOBLIST-1 end
```

```
LABO-1-Seq
    read LL
    LABO-1-BODY itr
        LABO-ACTION sel (DL-INTRO)
              DL-INTRODUIRE; write DP-INTRODUIRE to LP
              read LL
        LABO-ACTION alt (DA-INTRO)
              DA-INTRODUIRE; write DA-INTRODUIRE to LDL
              read LL
        LABO-ACTION alt (DA-ANN)
              DA-ANNULER; write DA-ANNULER to LDA
              read LL
        LABO-ACTION alt (DA-SUPPR)
              DA-SUPPRIMER; write DP-SUPPRIMER to LDL
              read LL
        LABO-ACTION alt (URG-VAL)
              URG-VALIDER; write DRG-VALIDER to LU
              read LL
        LABO-ACTION alt (DL-MDF)
              DL-MODIFIER; write DL MODIFIER to LDL
              read LL
        LABO-ACTION alt (DA-MDF)
              DA-MODIFIER: write DA-MODIFIER to LDA
              read LL
        LABO-ACTION alt (MED-ENRE)
              MED-ENRE; write MED-ENRE to LM
              read LL
        LABO-ACTION alt (MED-MDF)
              MED-MDF;  write MDF to LM
              read LL
        LABO-ACTION alt (A-ENRE)
              A-ENRE; write A-ENRE to LA
              read LL
        LABO-ACTION alt (A-MDF)
              A-MODIFIER; write A-MODIFIER to LA
              read LL
        LABO-ACTION alt (DEM-PRO-VER)
              DEM-PRO-VER; write DEM-PRO-VER to FV
              read LL
        LABO-ACTION alt (JOBLIST-DECL)
              JOBLIST-DECL; write JOBLIST-DECL to FJ
              read LL
        LABO-ACTION alt (TRANS-DECL)
              TRANS-DECL; write TRANS-DECL to FT
              read LL
        LABO-ACTION alt (HISADM-CHERCHER)
              HISADM-CHERCHER; write HISADM-CHERCHER to FMA
              read LL
        LABO-ACTION alt (HISMED-CHERCHER)
              HISMED-CHERCHER; write HISMED-CHERCHER to FHM
              read LL
        LABO-ACTION alt (PROTO-ED)
              PROTO-EDITOR; write PROTO-ED to FP
              read LL
        LABO-ACTION alt (DEM-PRO-VER)
              DEM-PRO-VER; write DEM-PRO-VER to LV
              read LL
        LABO-ACTION alt (DA-PRO-ACC)
              DA-PRO-ACC; write DA-PRO-ACC to LV
              read LL
        LABO-ACTION alt (DA-PRO-REF)
              DA-PRO-REG; write DA-PRO-REG to LV
              read LL
        LABO-ACTION end
    LABO-1-BODY end
LABO-1-Seq
```

```
LIT seq
        LIBERER
        read HL
        LIT-BODY itr
           LIT-ACTION seq
                ATTRIBUER;
                read HL
           LIT-ACTION seq
                LIBERER;
                read HL
           LIT-ACTION end
        LIT-BODY end
LIT end
```

```
MEDECIN-1 seq
        read L
        MED-ENRE
        read L
        MED-ACTION itr
          MED-MODIFIER
          read L
        MED-ACTION end
MEDECIN-1 end
```

```
PATIENT-1 seq
      read AP
      ENREGISTRER
      read AP&ST&SS&LP
      PATIENT-BODY itr
           PATIENT-ACTION sel (ATTRIBUER)
               ATTRIBUER, write ATTRIBUER to PH
               read AP&ST&SS&LP
           PATIENT-ACTION alt (LIBERER)
               LIBERER, write LIBERER to PH
               read AP&ST&SS&LP
           PATIENT-ACTION alt (DOS-OUVRIR)
               DOS-OUVRIR, write DOS-OUVRIR to PD
               read AP&ST&SS&LP
           PATIENT-ACTION alt (DOS-FERMER)
               DOS-FERMER, write DOS-FERMER to PD
               read AP&ST&SS&LP
           PATIENT-ACTION alt (DP-INTRODUIRE)
               DP-INTRODUIRE, write DP-INTRODUIRE to DPD
               read AD&SR&SS&LP
           PATIENT-ACTION alt (CONSOMMER)
                CONSOMMER, write CONSOMMER to PM
               read AP&ST&SS&LP
           PATIENT-ACTION alt (SIGN-MDF)
               SIGN-MODIFY
               read AP&ST&SS&LP
           PATIENT-ACTION alt (DL-INTRODUIRE)
               DL-INTRODUIRE, write DL-INTRODUIRE to PDL
               read AP&ST&SS&LP
           PATIENT-ACTION alt (TRANSFERER)
               TRANSFERER, write TRANSFERER to PH
               read AP&ST&SS&LP
           PATIENT-ACTION end
      PATIENT-BODY end
PATIENT-1 end
```

Rem : AP&ST&SS&LP est un "rough merge" de "data-stream"

```
SERV-SOINS-1 seq
     read S
     SS-BODY itr
          SS-ACTION sel (DP-INTRO)
               DP-INTRODUIRE; write DP-INTRODUIRE to ST
               read S
          SS-ACTION alt (DL-PRO-INTRODUIRE)
               DL-INTRODUIRE; write DL-PRO-INTRODUIRE to SP
               read S
          SS-ACTION alt (DA-PRO-INTRO)
               DA-PRO-INTRO; write DA-PRO-INTRO to SLP
               read S
          SS-ACTION alt (DA-PRO-ANN)
               DA-PRO-ANN; write DA-PRO-ANNULER to SAP
               read S
          SS-ACTION alt (ANT-INTRODUIRE)
               ANT-INTRODUIRE; write ANT-INTRODUIRE to SD
               read S
          SS-ACTION alt (FM-MDF)
               FM-MDF; write FM-MDF to SD
               read S
          SS-ACTION alt (CONSOMMER)
               CONSOMMER; write CONSOMMER to SP
               read S
          SS-ACTION alt (M-MDF)
               M-MDF; write M-MDF to SM
               read S
          SS-ACTION alt (RELATER)
               RELATER; write RELATER to SD
               read S
          SS-ACTION alt (CONCL-MDF)
               CONCL-MDF; write CONCL-MDF to SD
               read S
          SS-ACTION alt (L-REDIGER)
               L-REDIGER; write L-REDIGER to SD
               read S
          SS-ACTION alt (CLOTURER)
               CLOTURER; write CLOTURER to SD
               read S
          SS-ACTION end
     SS-BODY end
SERV-SOINS-1 end
```

```
SERV-TECH-1 seq
      read T
      ST-BODY itr
        ST-ACTION sel (DP-INTRO)
          DP-INTRODUIRE; write DP-INTRODUIRE to TDP
          read T
        ST-ACTION alt (DP-ANN)
          DP-ANNULER; write DP-ANNULER to TDP
          read T
        ST-ACTION alt (DP-MDF)
          DP-MODIFIER; write DP-MODIFIER to TDP
          read T
        ST-ACTION alt (CONSOMMER)
          CONSOMMER; write CONSOMMER to TP
          read T
        ST-ACTION alt (VALIDER)
          VALIDER; write VALIDER to TDP
          read T
        ST-ACTION alt (M-MDF)
          MODIFIER; write M-MDF to TM
          read T
        ST-ACTION end
      ST-BODY end
SERV-TECH-1 end
```

```
URG-ANAL-1 seq
        read AU&LU
        URG-INTRODUIRE
        read AU&LU
        URGA-ACTION itr (while URG-MODIFIER)
             URG-MODIFIER
             read AU&LU
        URGA-ACTION end
        URGA-TERM sel (AU)
             URG-ANNULER
        URGA-TERM alt (LU)
             URG-EVENT seq
                  URG-VALIDER
             URGENT-EVENT end
        URGA-TERM end
URG-ANAL-1 end
```

3.4. FUNCTION STEP

Il s'agit d'automatiser les fonctions reprises par les spécifications. Dans ce travail, ce sont spécialement les fonctions de laboratoire d'analyses médicales qui vont être développées. Ensuite, la fonction de consultation du dossier médical est abordée. L'impact de ces fonctions oblige le SSD à être remanié. Un nouveau SSD est construit.

3.4.1. Le Laboratoire d'analyses médicales - Fonctions

3.4.1.1. Les protocoles

Ils sont produits et édités tous les soirs. Cependant, dans un laboratoire, toutes les analyses ne sont pas effectuées tous les jours. Certaines sont faites deux fois par semaine, d'autres une fois par quinzaine, afin de regrouper les demandes du même type d'analyse qui nécessitent des produits coûteux pour la détermination du résultat. Un status des protocoles est défini en fonction de ces analyses.
Les analyses ont un type 0, si elles sont effectuées tous les jours, un type I, si elles sont effectuées deux fois par semaine (le même jour), un type II si elles sont effectuées une fois par quinzaine (le même jour).
Le protocole a un status Status-Prot "Partiel 0" si au moins une analyse journalière d'une demande n'a pas de résultat. Il a un Status-Prot "Partiel I" si au moins une analyse de type I n'a pas de résultat. Il a un Status-Prot "Partiel II " si au moins une analyse de type II n'a pas de résultat. Un protocole est complet si toutes les analyses demandées sur une demande de laboratoire ont un résultat non provisoire.

Le résultat d'une demande d'analyses est soit "à blanc" (pas de résultat), soit un résultat provisoire, soit un résultat à vérifier, soit un résultat définitif.
Il est fréquent dans un laboratoire que les résultats soient modifiés. Normalement, en indiquant un résultat mais aussi par erreur d'encodage, par passage d'un résultat à vérifier à un résultat définitif...
N.B. Un résultat à vérifier n'est pas imprimé sur le protocole (il est stipulé -en cours-). Un résultat provisoire est imprimé sur le protocole avec la mention qu'il est provisoire.

Il est aussi évident qu'un protocole d'un type quelconque déjà imprimé mais dont un résultat est modifié par la suite (par exemple : le résultat définitif est changé ou erreur d'encodage) devra être réimprimé.
Un "STATUS-IMPRIM" existe conjointement à "STATUS-PROT" pour chaque demande.
La validation du protocole se fait lors de la signature du médecin sur le protocole.
Si le médecin n'est pas satisfait, il modifie un ou plusieurs résultats.
Le protocole a un certain "layout" qui est défini. Les demandes d'analyses sont placées dans un certain ordre et des commentaires peuvent être ajoutés.

a) FVERCO : Cette fonction vérifie la cohérence des statuts de la demande du laboratoire chaque fois qu'un résultat d'une demande d'analyse est modifié ou annulé.

Une manière de procéder (parmi d'autres)

- Chercher toutes les demandes d'analyse de la demande de laboratoire dont une demande d'analyse a été modifiée ou annulée
- Déterminer quel est le nouveau STATUS-PROT
- Mettre à jour le nouveau STATUS-PROT et mettre le STATUS-IMPRIM à NON-IMPRIME.

Fonction interactive.

```
FVERCO itr
        read LFV (dem-analyse-id)
        getsv FVLSV (dem-analyse-id)
        dem-lab = dem-lab-id
        i := 0
        RECHERCHE-DEM-ANALYSE itr
              getsv FVASV (dem-lab)
              dem-an := dem-analyse-id
              caba  i := dem-an
              i :=  1
        RECHERCHE-DEM-ANALYSE end
        Status-prot := determ-status-prot-dl (tab      )
        status-imprim := "non imprimé"
        write DL-STATUS-MDF to FV
FVERCO end
```

Un seul processus peut assurer la vérification de toutes les modifications.

Accès nécessaires

DBFV1 getsv FVLSV (dem-analyse-id) :
accès à la demande de laboratoire à laquelle appartient la demande d'analyse spécifiée comme argument

DBFV2 getsv FVASV (dem-labo-id)
accès à une demande d'analyse associée à la demande de laboratoire spécifiée comme argument

Fonction à développer

FBN1 déterm-status-prot-dl( )
Cette fonction détermine le status d'une demande de laboratoire en recevant comme argument toutes les demandes d'analyse qu'elle contient.

Rem : Le fait de mettre un status-prot dans la demande de laboratoire est déjà plutôt une optimisation et peut être considéré en partie comme une décision d'implémentation. Dans le cas où cette variable n'est pas incluse dans le demande de laboratoire, il faut vérifier le status du protocole par une fonction qui inspecte les vecteurs d'état de tous les processus des demandes d'analyses associées à cette même demande. Quant au status-imprim, il pourrait être remplacé par une action IMPRIMER. Cependant, étant donné que ces deux status sont importants et liés intimement, il semblerait plus logique d'associer une action DL-STATUS-MODIFIER (Conjointement, cette fonction permet de garder le status de la demande sans devoir accéder à toutes les demandes d'analyse de celle-ci.

b) FPROTO : Les protocoles sont construits (et édités) après activation de cette fonction. Cette fonction ne génère que des protocoles qui ne sont pas encore imprimés (status-imprim = non-imprimé).

Manière de procéder

- Changer toutes les caractéristiques des analyses qui sont nécessaires à l'élaboration du protocole
- Déterminer toutes les demandes de laboratoire à imprimer
- Pour chaque demande de laboratoire sélectionnée :
  - déterminer toutes les demandes d'analyses
  - chercher la signalétique du patient et du médecin associé à cette demande
  - élaborer le protocole
  - modifier le status-imprim de la demande de laboratoire en le mettant à "imprimé"

Fonction interactive.

```
FPROTO seq
        read FP
        PROTO-EDITER
        i := 0
        LOAD-ANAL-DS itr
                getsv FP SV
                tabds i,...  := (analds, ...)
                i := i + 1
        LOAD-ANAL-DS end
        CONSTRUCT-PROT itr
                getsv FPLSV (status-imprim = "non-imprimé")
                dem-lab := dem-lab-id
                i := 0
                SEARCH-DEMANDE-ANALYSE itr
                    getsv FPASV (dem-lab)
                    dem-ana := dem-ana-id
                    i := i + 1
                SEARCH-DEMANDE-ANALYSE end
                getsv FPMSV (dem-lab)
                med-prescr := med-prescr-id
                getsv FPPATSV (dem-lab)
                pat := pat-id
                produire-protocole (tabds, dem-lab, dem-ana, med-prescr, pat)
                status-impr := "imprimé"
                write DL-STATUS-MDF to FPL
        CONSTRUCT-PROT end
FPROTO end
```

Un seul processus peut assurer la production des protocoles.

Accès nécessaires

DBFP1 getsv FPDSV
accès (séquentiel) à un descripteur d'une analyse

DBFP2 getsv FPLSV (status-imprim = non-imprimé)
accès à une demande de laboratoire dont le status imprim est "non-imprimé"

DBFP3 getsv FPASV (dem-lab-id)
accès à une demande d'analyse associée à une demande de laboratoire spécifié comme argument

DBFP4 getsv FPMSV (dem-lab-id)
accès au médecin prescripteur de la demande de laboratoire spécifié comme argument

DBFP5 getsv FPPATSV (dem-lab-id)
accès à la signalétique du patient pour lequel la demande de laboratoire, spécifiée comme argument, est prescrite

3.4.1.2. Les analyses urgentes

Certaines analyses sont à faire en urgence. Les résultats doivent de ce fait être validés et transmis le plus vite possible.
Rem : C'est à l'"input" qu'est vérifié si l'analyse peut ou non être effectuée en urgence.

a) FURG : Cette fonction s'occupe d'accorder l'entité URG-ANA avec les demandes d'analyses (DA-EX).

Manière de procéder

- Lors de l'introduction d'une demande d'analyse :
  - si la demande d'analyse est urgente, introduite une analyse urgente

- Lors de la modification d'une demande d'analyses
  - si la demande d'analyse est modifiée de telle façon que d'urgente elle passe en non-urgente, annuler l'analyse urgente correspondante.
  - si la demande d'analyse est modifiée de telle façon que de non urgente elle passe à urgente, introduite une nouvelle analyse urgente
  - si la demande d'analyse est modifiée de telle façon qu'elle reste urgente et que les résultats sont modifiés, modifier l'analyse urgente correspondante

- Lors de l'annulation d'une demande d'analyse :
  - si la demande d'analyse est urgente, annuler l'analyse urgente correspondante.

Fonction "embedded" dans DA-EX.

```
DA-EX-1 seq
     .
     .
     .
     DA-INTRO;
     if (urgent-flag = 'oui') write URG-INTRO to DAU
          .
          .
          .
          DA-MDF;
          if (urgent-flag : 'oui') et (urgent flag (dem-ana-mdf) = 'non')
               getsv FUSV (dem-anal-id)
               if (urg-anal not VALIDEE) write URG-ANN to DAU
          if (urgent flag = 'oui' ) et (urgent flag (dem-anal-mdf) = 'oui')
               write URG-INTRO to DAU
          if (urgent flag = 'oui') et (urgent flag (dem-anal-mdf) = 'oui')
               write URG-MDF to DAU
               .
               .
               .
     DA-ANNULER;
     if (urgent flag = 'oui'), write URG-ANN to DAU
     .
     .
     .
DA-EX-1 end
```

Un processus est associé à chaque processus de DA-EX. Il ne contient cependant aucune variable.

Accès nécessaires

DBFU1 getsv FUSV (dem-anal-id)
accès à l'analyse urgente, associée à la demande d'analyse spécifiée comme argument

b) FTRANSM : Fonction qui transmet les résultats des demandes d'analyses urgentes après que le médécin les ait validés chacun

Manière de procéder

- Rechercher toutes les analyses urgentes validées par service

- Pour chaque analyse urgente validée :
  - chercher la signalétique du patient
  - chercher toutes les autres analyses urgentes relatives à un même patient
  - introduire l'action "transmettre" dans chacun des processus pris en compte

- Transmettre toutes les analyses vers les différents services

Fonction interactive.

```
FTRANSM seq
    read FT
    TRANS-DECLENCHER;
    TRANS-ANAL itr
        getsv FTUSV (VALIDEE)
        urg-ana := urg-ana-id
        getsv FTPSV (urg-ana)
        dem-lab := dem-lab-id
        i := 0
        SEARCH-ANAL-URG itr
            getsv FTUSV (VALIDEE) dem-lab
            urg-ana := urg-ana-id
            labo (i) := urg-ana
            i := i + 1
            write URG-TRANS to PTU
        SEARCH-ANAL-URG end
    TRANS-ANAL end
    sort-and-print-to-service ()
FTRANSM end
```

"Function process" unique.

Accès nécessaires

DBFT1 getsv FTUSV (VALIDE)
accès à une analyse urgente qui est déjà validée

DBFT 2 getsv FTPSV (urg-ana-id)
accès à la demande de laboratoire associée à l'analyse urgente, spécifié comme argument

DBFT3 getsv FTUSV (VALIDE, Pat-id)
accès à une analyse urgente qui est déjà validée appartenant à la demande de laboratoire spécifiée comme second argument

Fonction à développer

FBN2 sort-and-print-to-service()
Cette fonction trie les analyses urgentes validées par service, par demande de laboratoire et se charge de les envoyer vers les services correspondants.

3.4.1.3. Les joblistes et autres listes

Le but des joblistes est de faciliter le travail des techniciens dans le laboratoire. Les analyses qui sont à effectuer sortent sur un listing, triées par secteur. Les demandes sont imprimées par ordre croissant des numéros. Sur les différentes joblistes (une par secteur), toutes les demandes de laboratoire non complètes sont reprises avec leur résultat pour les demandes d'analyses déjà répondues. Une spécification supplémentaire précise que l'on désire les résultats des mêmes analyses antérieures si elles existent sur les joblistes correspondantes.

a) FJOBLIST : Production des joblistes pour le laboratoire

Manière de procéder

- Les joblistes précédentes sont annulées

- Recherche de toutes les demandes de laboratoire incomplètes

- Pour chacune de ces demandes :
  - rechercher toutes les demandes d'analyse et les mettre dans un tableau (tableau analyse, 0)
  - rechercher le patient associé à cette demande de laboratoire
  - rechercher toutes les demandes de laboratoire antérieures associées à ce patient

Pour chacune de ces demandes de laboratoire (chronologiquement)

- Rechercher toutes les demandes d'analyses
- Si une ou plusieurs des analyses demandées (tableau analyse, 0) existent dans les demandes antérieures de laboratoire, les mettre dans le tableau (tableau analyse, n)
- Trier le tableau et formater les joblistes

"Function imposed" sur les demandes de laboratoire et d'analyses mais interactive sur joblistes.

```
FJOBLIST seq
    read FJ
    JOBLIST-DECLENCHER
    ERASE-JOBLIST itr.
        getsv FJJLSV (ETABLIE)
        write JL-SUPPRIMER to FJL2
    ERASE-JOBLIST end
    SEARCH-DEM-LAB itr
        getsv FJLSV (not COMPLET)
        dem-lab := dem-lab-id
        getsv FJPATSV (dem-lab)
        pat := pat-id
        SEARCH-DEM-ANALYSE itr
            getsv FJASV (dem-lab)
            tabanal (anal, 0) := anal-id
        SEARCH-DEM-ANALYSE end
        SEARCH-DEM-PREVIOUS itr
            n := 1
            getsv FJLSV (pat)
            dem-lab := dem-lab-id
                SEARCH-ANAL-PREVIOUS itr
                    getsv FJASV (dem-lab)
                    dem-anal in tableau ( )print to tabanal (..,n)
                SEARCH-ANAL-PREVIOUS end
            n := n + 1
        SEARCH-DEM-PREVIOUS end
    SEARCH-DEM-LAB end
    sort-and-format (tabanal (...,...))
    PRINT-JOBLIST itr
        write JL-ETABLIR to FJL
    PRINT JOBLIST end
FJOBLIST end
```

Un seul processus pour produire les joblistes.

Accès nécessaires

DBFJ1 getsv FJJLSV (ETABLI)
accès à une jobliste établie

DBFJ2 getsv FJLSV (not COMPLET)
accès à une demande de laboratoire non complète

DBFJ3 getsv FJPATSV (dem lab)
accès au patient associé à une demande de laboratoire spécifié comme argument

DBFJ4 getsv FJASV (dem-lab)
accès à une demande d'analyse associée à une demande de laboratoire spécifié comme argument

DBFJ5 getsv FJLSV (pat)
accès à une demande de laboratoire associée à un patient spécifié comme argument

Fonction à développer

FBN3 sort-and-format()
Cette fonction trie toutes les analyses par jobliste respective et les formate pour établir une nouvelle occurence de "JOBLIST" qui, en les imprimant, servira au technicien du laboratoire mais aussi à l'input pour l'introduction des résultats par joblist.

b) FHISADM : Cette fonction détermine toutes les demandes de laboratoire d'un patient, à partir de son identifiant.

Manière de procéder

- Rechercher toutes les demandes antérieures des laboratoires pour un patient donné

Function imposed

```
FHISADM seq
     read FHA (pat)
     HISADM-CHERCHER;
     SEARCH-DEM-LABO itr
          getsv FHALSV (pat)
          print dem-lab-id to
     SEARCH-DEM-LABO end
FHISADM end
```

Un seul processus.

Accès nécessaire

DBFHA1 getsv FHALSV (pat)
accès à une demande de laboratoire d'un patient spécifié comme argument.

c) FHISMED : Cette fonction détermine toutes les demandes d'analyses au travers des différentes demandes de laboratoire, pour une ou plusieurs analyses fonction "imposed".

Manière de procéder

- Rechercher toutes les demandes antérieures de laboratoire relatives à un patient

- Pour chaque demande antérieure de laboratoire (chronologiquement)
  - chercher toutes les demandes d'analyse
  - pour chacune de ces demandes d'analyse :
    . si une demande d'analyse est identique à une analyse reprise dans le tableau d'analyse passé comme argument, il faut le mettre dans une liste

Function imposed.

```
FHISMED seq
     read FHM (Pat     , code-anal ( ) )
     HISMED-CHERCHER;
     SEARCH-DEM-LABO itr
          getsv FHMLSV (pat  )
          dem-lab := dem-lab-id
          SEARCH-DEM-ANAL itr
               getsv FHMASV (dem-lab)
               dem-analyse := dem-analyse-id
               if (dem-analyse in code-analyse ( ) ), print to list
          SEARCH-DEM-ANAL end
     SEARCH-DEM-LABO end
FHISMED end
```

Processus unique

Accès nécessaires

DBFHM1 getsv FHMLSV (pat)
accès à une demande de laboratoire associé à un patient spécifié comme argument

DBFHM2 getsv FHMASV (dem-lab)
accès à une demande d'analyse associé à une demande de laboratoire

3.4.1.4. Les demandes provisoires

Les demandes provisoires sont encodées dans les services de soins par le personnel la veille (au soir) de leur prélèvement. Les échantillons sont acheminés au laboratoire avec une demande et un numéro. Les encodeuses n'ont plus qu'à vérifier que les demandes d'analyses sont correctes et qu'il ne manque pas de sang.

FVDEMPRO : Fonction qui sert à vérifier la demande provisoire et à les introduire comme demande (définitive)

Manière de procéder

- La fonction reçoit en entrée un identifiant d'une demande provisoire qui est recherché dans la base de données
- Si la demande existe (est introduite), toutes les demandes provisoires d'analyse sont recherchées
- Une procédure les affiche toutes et permet à l'utilisateur de répondre par "annuler" ou "accepter"
- Les réponses sont lues pour toutes les demandes d'analyse et transmises aux entités correspondantes
- Si aucune analyse n'est acceptée, alors la demande de laboratoire est annulée, sinon elle est acceptée; la demande de laboratoire et toutes les demandes d'analyse sont renvoyées dans le système comme demandes définitives.

Fonction interactive

```
FVDEMPRO seq
    read LV
    DEM-PRO-VER
    getsv DLPSV (dem-pro)
    VER-D-BODY sel (DL-PRO-INTRODUITE)
        VER-D-ACTION seq
            VER-A-BODY itr
                getsv DAPSV (DA-PRO-INTRODUITE, dem-l-pro)
            VER-A-BODY end
            afficher-et-choisir ()
        VER-D-ACTION seq
            REP-A-BODY itr
                read LV
                REP-A-EVENT sel (DA-PRO-ACC)
                    DA-PRO-ACC; write DA-PRO-ACC to FDAP
                REP-A-EVENT alt (DA-PRO-REF)
                    DA-PRO-REF; write DA-PRO-REF to FDAP
                REP-A-EVENT end
            REP-A-BODY end
        VER-D-ACTION seq
            CLOTURE BODY sel (aucunes analyses acceptées)
                DL-PRO-REF; write DL-PRO-REF to FDLP
            CLOTURE-BODY alt (else)
                ACCEPTATION-ANA seq
                    DL-INTRO; write DL-INTRO to VL
                    INTRO-ANA itr, while (DA-PRO-ACCEPTEE)
                        DA-INTRO; write DA-INTRO to VL
                    INTRO-ANAL end
                ACCEPTATION-ANA end
            CLOTURE-BODY end
        VER-D-ACTION alt (else)
    VER-D-BODY end
FVDEMPRO end
```

Processus unique

Accès nécessaires

DBFVDP1 getsv DLPSV (dem-l-pro)
accès à la demande provisoire spécifiée comme argument

DBFVDP2 getsv DAPSV (DA-PRO-INTRODUIRE, dem-l-pro)
accès à une demande provisoire d'analyse "introduite" associé à une demande de laboratoire spécifié comme second argument

Fonction à développer

FBN4 Afficher-et-choisir.
Cette fonction affiche toute la demande d'analyse provisoire et permet à l'utilisateur d'introduire pour chaque demande, la réponse qu'il souhaite. Les réponses sont bien sur réintroduites par l'input-labo (read LV).

3.4.2. Consultation du dossier médical

Dans ce développement, la notion de dossier médical n'est pas présente. Tous les éléments du dossier médical sont présents mais répartis entre ELEM-DOS, DL-EX et DA-EX, DP-EX. La fonction de consultation a comme but d'accéder à ces différentes parties et de les ordonner. L'ordre chronologique semble le plus souhaitable.

Le problème du dossier médical est compliqué. Le plus difficile cependant est d'établir un consensus sur sa structure. Les actions précisées dans ELEM-DOS en ce qui concerne le dossier lui-même (ant-intro, relater, lettre-rediger) ne sont certes pas complètes. Une analyse plus poussée du contexte du dossier est mis dans l'annexe B. A ce premier écueil, un second s'ajoute, l'interface médecin-machine difficile à automatiser pour qu'elle soit utilisée.

Manière de procéder

- Vérifier si l'utilisateur a accès aux données du patient
- Chercher pour un patient donné
  - tous les éléments dos et vérifier si l'accès est permis
  - toutes les demandes de laboratoire associées et vérifier si l'accès est permis
  - pour chaque demande de laboratoire, toutes les demandes d'analyse et vérifier si l'accès est permis
  - toutes les demandes de protocole et vérifier si l'accès est permis
- Les "merger" chronologiquement

3.4.3. Tarification

L'option a été prise de centraliser tous les actes à tarifier et facturer. Ceux-ci seront effectivement tarifiés et facturés par un processus extérieur à ce modèle.

La fonction FCOMPTA est chargée de produire le fichier intermédiaire. C'est une simple fonction interactive sur MEDIC,DP-EX, DL-EX, CELEM-DOS, CHOSPITALISATION.

Dans ces deux derniers types d'entités, elle se charge de comptabiliser même partiellement les prestations à facturer. En effet, un patient qui reste trois mois, par exemple, à l'hôpital sous la responsabilité du même médecin, doit pouvoir être facturé régulièrement (spécification).

Le schéma est suffisamment clair. A noter que cette fonction oblige le dédoublement des entités en un level-2.

3.4.4. Interface Modèle/Monde réel pour le laboratoire

Un schéma est associé à chacune des fonctions proposée pour l'encodage des laboratoires (INPUT-LABO). Ceci est un scénario possible parmi plusieurs. Avec chaque fonction d'input, des nouvelles actions ont été précisées. Elles devront permettre de comprendre la séquence décidée, mais surtout de programmer les fonctions de bas niveau manquantes (qui dépendent plus de la machine, du langage, etc...). Il est assez aisé de déduire une suite de menus pour cet interface.

3.4.4.1. Introduction et mise a jour des analyses

Le même type de diagramme que celui utilisé par les autres processus est employé pour représenter une façon de réaliser la gestion des analyses.
cfr schéma.

ACTION 1 :
L'action 1 correspond à une introduction du code de l'analyse à l'écran. Si l'analyse existe, elle est affichée, sinon des champs par défaut sont affichés.

ACTION 2 :
L'action 2 permet de modifier les champs. Suivant l'existence de l'analyse ou non, elle est A-MODIFIER ou A-ENREGISTRER.

Accès nécessaires

DBIL1 getsv ANALSV (analds-id)
accès à une description d'analyse à partir d'un identifiant spécifié comme argument

3.4.4.2. Validation des analyses urgentes

cfr schéma

ACTION 1 :
Validation à l'écran des analyses affichées

Accès nécessaires

DBIL2 getsv URGASV
accès à une analyse urgente

3.4.4.3. Encodage des demandes d'analyse

cfr schéma

ACTION 1 :
Introduction du numéro de demande/nodossier (pat-id)

ACTION 2 :
Introduction du MEDECIN

ACTION 3 :
Introduction de la ligne d'analyse

FVALID :
Dissection de la ligne et validation des analyses et codage des analyses

Accès nécessaires

DBIL3 getsv PATSV (pat-id)
accès à la signalétique patient, à partir de l'identifiant spécifié (nodossier)

DBIL4 getsv MEDSV (med-id)
accès à la signalétique médecin à partir d'un identifiant (le numéro interne à l'établissement)

DBIL5 getsv DLSV (n° demande)
accès à une demande de laboratoire associée à ce numéro de demande

DBIL! getsv ANALSV (analds-id)
accès identique à DBIL1
rem : utilisé par FVALID

3.4.4.4. Modification de la demande de laboratoire et d'analyse

(sauf suppression de la demande de laboratoire)
cfr schéma

ACTION 1 :
Introduction du numéro de demande

ACTION 2 :
Introduction des modifications apportées à la signalétique médecin

ACTION 3 :
Affichage des médecins + choix de l'un d'eux

ACTION 4 :
Introduction des modifications à la demande du laboratoire

ACTION 5 :
Validation de ces modifications

ACTION 6 :
Transformation des codes-analyses en mnémoniqueet affichage des demandes d'analyses

ACTION 7 :
Introduction des modifications (ligne-analyse)

FVALID :
Dissection de la ligne, validation et codage des analyses

Accès nécessaires

DBIL! getsv DLSV (n° demande)
identique à DBIL5

DBIL! getsv MEDSV (med-id)
identique à DBIL4

DBIL6 getsv DASV (n° demande)
accès à une demande d'analyse associé à une demande de laboratoire spécifié comme argument

DBIL! getsv ANALSV (analds-id)
identique à DBIL1

3.4.4.5. Encodage résultats

cfr schéma

ACTION 1 :
Introduction des résultats pour toutes les demandes d'analyse d'une demande de laboratoire

ACTION 2 :
Introduction des résultats par type d'analyse pour toutes les demandes de laboratoire en cours

ACTION 3 :
Introduction des résultats par joblistes

Accès nécessaires

DBIL! getsv DASV (n° demande)
accès identique à DBIL6

DBIL7 getsv DASV (not REPONDU, analyse-id)
accès à toutes les demandes d'analyses non répondues d'un même type d'analyse spécifié comme second argument
rem : REPONDU peut être spécifié comme résultat à blanc et résultat à vérifier, par exemple

DBIL8 getsv JLSV (jl-numéro)
accès à une jobliste spécifié comme argument

3.4.5. Validation des actions (Input subsystem)

Pour chaque action que l'on introduit, et qui reflète le monde réel, les erreurs ne sont pas possible (dans le cas où le modèle est correct).
Cependant, l'utilisateur peut ne pas avoir introduit les actions à temps ou tout simplement se tromper. Pour éviter de perturber le déroulement des différents processus, au niveau de l'input system, un processus (avant de laisser pénétrer les actions dans le modèle (le système)), contrôle que l'occurence de telle entité peut la supporter. Pour cela, des conditions testent pour l'occurence un type d'entité où se trouve le "text-pointer".

Cette validation se fait entre le point d'entrée dans le modèle et le processus de distribution des actions aux différents processus (le "scheduler" en quelque sorte), appelés ici INPUT-LABO, INPUT-ADM, ...
Elle n'est plus représentée dans le JSD et est considérée comme incluse dans l'INPUT-XXX afin de ne pas surcharger les schémas inutilement.

Une table de condition permet de déterminer quelles valeurs le "text-pointer" doit satisfaire.

1) Pour le laboratoire

```
LAB-CONTROL itr
     read L
     CONTROL-BODY sel (DL-INTRO)
          CONTROL-SUBBODY sel, cond-dlintro
               write DL-INTRO to LL
          CONTROL-SUBBODY alt (else)
               print err-dlintro
          CONTROL-SUBBODY end
     CONTROL-BODY alt (DA-INTRO)
          CONTROL-BODY sel, cond-daintro
               write DA-INTRO to LL
          CONTROL SUBBODY alt (else)
               print err-dlintro
          CONTROL-SUBBODY end
     CONTROL-BODY alt (DA-ANN)
          CONTROL-SUBBODY sel, cond-daann
               write DA-ANN to LL
          CONTROL-SUBBODY alt (else)
               print err-daann
          CONTROL-SUBBODY end
     CONTROL-BODY alt (DA-SUPPR)
          CONTROL-SUBBODY ....
                    .
                    .
                    .
                    .
                    .

     CONTROL-BODY end
LAB-CONTROL end
```

2) Pour le service soins

```
SERVS-CONTROL itr
     read S
     CONTROL-BODY sel (DL-PRO-INTRO)
          CONTROL-SUBBODY sel cond-dlprointro
               write DL-PRO-INTRO to SS
          CONTROL-SUBBODY alt (else)
               print err-dlprointro
          CONTROL-SUBBODY end
     CONTROL-BODY alt (DA-PRO-INTRO)
          CONTROL-SUBBODY alt cond-daprointro
               write DA-PRO-INTRO to SS
          CONTROL-SUBBODY alt (else)
               print err-daprointro
          CONTROL-SUBBODY end
     CONTROL-BODY alt (DA-PRO-ANN)
          CONTROL-SUBBODY alt (cond daproann)
               write DA-PRO-ANN to SS
          CONTROL-SUBBODY alt (else)
               print err-daproann)
          CONTROL-SUBBODY end
     CONTROL-BODY end
SERVS-CONTROL end
```

3) Pour l'administration

```
ADM-CONTROL itr
     read A
     CONTROL-BODY sel (ENREGISTRER)
          CONTROL-BODY sel (cond-enre)
               write ENREGISTRER to AA
          CONTROL-SUBBODY alt (else)
               print err-enregistrer
          CONTROL-SUBBODY end
     CONTROL-BODY alt (ATTRIBUER)
          CONTROL-SUBBODY sel (cond-attribuer)
               write ATTRIBUER to AA
          CONTROL-SUBBODY alt (else)
               print err-attribuer
          CONTROL-SUBBODY end
     CONTROL-BODY alt (TRANSFERER)
          CONTROL-SUBBODY sel (cond-transferer)
               write TRANSFERER to AA
          CONTROL-SUBBODY alt (else)
               print err-transferer
          .
          .
          .
     CONTROL-BODY end
ADM-CONTROL end
```

3.4.6. Modification du SSD

Les processus de Level-2 modifient le SSD. Un nouveau SSD est construit.

NOUVEAUX TYPES D'ENTITE :
- DP-EX-2
- DL-EX-2
- CHOSP-2
- CELEM-DOS-2
- MEDICAMENT-2
- JOBLIST-2

DL-EX de toute façon allait générer un level-2 à cause des fonctions FPROTO et FVERCO qui sont aussi interactives :
- DA-EX-2

DA-EX se dédouble à cause de l'inclusion de la fonction FURG qui introduit, modifie et annule les analyses urgentes :
- URG-ANA-2

URG-ANA est connecté à une fonction interactive FTRANS et donc nécessite un nouveau type level-2 :
- DEM-L-PRO-2
- DEM-A-PRO-2

La fonction FVERDEMPRO est interactive sur les deux entités donc un nouveau processus level-2 est nécessaire.

Les processus de level-1 qui deviennent des processus type "scheduler" comme LABO, ADMINISTRATION, SERV-SOINS, ne sont pas recopiés.

```
DA-EX-2 seq
     read DA2
     DA-INTRODUIRE
     read DA2
     DA-ACTION itr
          DA-MODIFIER
          read DA2
     DA-ACTION end
     DA-ANNULER
DA-EX-2 end
```

```
DL-EX-2 seq
     read DL2&FPL&FVL&FCL
     DL-INTRODUIRE
     read DL2&FPL&FVL&FCL
     DL-BODY itr
          DL-ACTION sel (DA-INTRO)
               DA-INTRODUIRE; write DA-INTRODUIRE to LA
               read DL2&FPL&FVL&FCL
          DL-ACTION alt (DL-MDF)
               DL-MODIFIER; write DL-MODIFIER to LA
               read DL2&FPL&FVL&FCL
          DL-ACTION alt (DL-ST-MDF)
               DL-STATUS-MODIFIER
               read DL2&FPL&FVL&FCL
          DL-ACTION end
     DL-BODY end
     DL-TERMINAISON sel (DL-SUPPRIMER)
          DL-SUPPRIMER
     DL-TERMINAISON alt (DL-COMPTAB)
          DL-COMPTABILISER
     DL-TERMINAISON end
DL-EX-2 end
```

```
CELEM-DOS-2 seq
     read CCD&FCD
     DOS-OUVRIR
     read CCD&FCD
     DOS-TARIF itr, while (not DOS-FERMER)
          DOS-PRECOMPTABILISER
          read CCD&FCD
     DOS-TARIF end
     DOS-FERMER
     read CCD&FCD
     DOS-COMPTABILISER
CELEM-DOS-2 end
```

```
CHOSPITALISATION-2-seq
     read CCH&FCH
     ATTRIBUER
     read CCH&FCH
     H-TARIF itr, while (not LIBERER)
          H-PRECOMPTABILISER
          read CCH&FCH
     H-TARIF end
     LIBERER
     read CCH&FCH
     H-COMPTABILISER
CHOSPITALISATION-2 end
```

```
DEM-A-PRO-2 seq
     read AAP&FDAP
     DA-PRO-INTRO
     read  AAP&FDAP
          DEM-A-BODY sel (DA-PPO-ACC)
               DA-PRO-ACC
          DEM-A-BODY alt (DA-PRO-REF
               DA-PRO-REF
          DEM-A-BODY alt (DA-PRO-ANN)
               DA-PRO-ANN
          DEM-A-BODY end
DEM-A-PRO-2 end
```

```
DEM-L-PRO-2 seq
     read LLP&DLP
     DL-PRO-INTRO
     read LLP&DLP
     INTRO-ANAL itr
          DA-PRO-INTRO; write DA-PRO-INTRO to LAP
          read LLP&DLP
     INTRO-ANAL end
     DEM-L-BODY sel (DA-PRO-ACC)
          DL-PRO-ACC
     DEM-L-BODY alt (DL-PRO-REF)
          DL-PRO-REF
     DEM-L-BODY alt (DL-PRO-ANN)
          DL-PRO-ANN
     DEM-L-BODY end
DEM-L-PRO-2 end
```

```
D-MEDICAMENT-2 seq
     read MM2&FCM
     CONSOMMER
     read MM2&FCM
     M-ACTION itr
          M-MODIFIER
          read MM2&FCM
     M-ACTION end
     M-COMPTABILISER
D-MEDICAMENT-2 end



DP-EX-2 seq
     read DP2
     DP-INTRODUIRE
     read DP2
     DP-BODY sel (DP-ANNULER)
          DP-ANNULER
     DP-BODY alt (else)
          DP-ACTION seq
               RESULTAT itr
               DP-MDF
               read DP2
               RESULTAT end
               DP-VALIDER
               read FCP
               DP-COMPTABILISER
          DP-ACTION end
     DP-BODY end
DP-EX-2 end

JOBLIST-2 seq
     read FJL
     JL-ETABLIR
     read FJL2
     JL-SUPPRIMER
JOBLIST-2 end


URG-ANAL-2 seq
     read UU
     URG-INTRODUIRE
     read UU
     URGA-ACTION itr, while (URG-MODIFIER)
          URG-MODIFIER
          read UU
     URGA-ACTION end
     URGA-TERM sel (URG-ANNULER)
          URG-ANNULER
     URGA-TERM alt (else)
          URG-EVENT seq
               URG-VALIDER
               read FTU
          URG-EVENT seq
               URG-TRANSMETTRE
          URG-EVENT end
     URGA-TERM end
URG-ANAL-2 end
```

3.5. SYSTEM TIMING STEP

Cette étape n'est pas développée dans cet exemple. Elle s'adresse essentiellement aux processus temps réel.

3.6. IMPLEMENTATION STEP

1. Décisions d'implémentation

Après avoir décrit les fonctions, il faut passer à la phase d'implémentation. Il faut transformer un processus en de multiples processus concurrents. Pour cela, un seul texte de programme est établi et un vecteur d'état associés à chaque occurence d'un type d'entité. Un pointeur de texte est associé à chaque occurence qui spécifie où dans la séquence des actions déterminées dans les étapes précédentes, celle-ci se trouve.

Le SID (I) peut maintenant être construit. Les fonctions sont difficiles à représenter, de même que les mémoires secondaires, si l'on ne veut surcharger le dessin. Il est décidé d'implémenter le projet sur plusieurs processeurs logiques. Les différents processeurs sont répartis comme suit :
- un processeur par service de soins
- un processeur par service d'examens techniques
- un processeur pour le laboratoire
- un processeur pour l'administration (et le processus médecin)

Après restructuration du SID, on peut démembrer les processus qui sont à cheval sur deux processeurs. La technique du démembrement permet à un processus dont des actions différentes sont transmises toujours par le même data-stream d'être séparé en deux parties (ou plusieurs). L'unité logique reste entière au niveau du type d'entité bien que le texte du processus soit éclaté en deux ou plusieurs morceaux. Les trois processus démembrés sont montrés ci-après (DEM-A-PRO, DEM-L-PRO, DL-EX). Le processus médecin ne peut cependant être démembré pour la simple et unique raison que ce sont les mêmes actions qui sont communiquées de part et d'autre par les processus qui lui sont reliés. On obtient un nouveau SID (II). Comme dans le cas du projet, c'est surtout le laboratoire d'analyses médicales qui est le centre d'intérêt, le SID (III) est éclaté en trois sous-schémas plus détaillés :
- le processus de laboratoire,
- celui de l'administration (et du médecin)
- celui des services de soins (gestion des demandes provisoires).

Ceci permet déjà de commencer la réalisation du système d'information au niveau de ces trois "services". Ici, le stockage des "state-vector" est ajouté.

2. Elaboration de la base de données

A la fin de l'étape 6, le "design" de la base de données doit être fait. Il faut considérer les options prises pour l'implémentation. Il faut considérer la base de données comme une collection de vecteurs d'état. Dans ce cas-ci, deux processeurs "logiques" (le processus médecin sera adjacent à celui de l'administration) sont prévus (les données relatives à DEM-X-PRO sont situées sur le processeur du labo). Une ébauche de base de données est dans l'annexe. On voit que la base de données est répartie. Suivant le type de base de données utilisé, les transformations vues au cours de Mr. Hainaut restent tout-à-fait d'application (une variable text-pointer est à rajouter à chaque vecteur d'état). (Dans certains cas, les données peuvent être démembrées comme les processus).

Rem 1 : La fonction d'archivage

Il est intéressant de noter que certaines actions et fonctions dépendent de l'étape d'implémentation plutôt que de l'étape fonction. La fonction FVERCO a déjà soulevé ce problème plus haut. L'archivage est uniquement une façon différente de stocker les vecteurs d'état. Le modèle et les fonctions ne sont pas impliqués par cette fonction. C'est un problème qui concerne uniquement le système que l'on développe.

Les données sont doubles : les données médicales et les données administratives. Les données administratives ne présentent aucun problème pour la raison toute simple qu'une fois "comptabilisées" (c'est-à-dire qu'un fichier reprenant tous les items à facturer - médicament, prestations, actes techniques - a été généré), elles ne nécessitent que rarement leur mise dans le réel.
Les données médicales posent plusieurs problèmes. La fiche médicale reste on-line dans la partie administrative du système. Les autres éléments qui constituent le dossier médical (ELEM-DOS, DP-EX, DL-EX, DA-EX) ne peuvent pas être archivés tant que le dossier n'est pas clôturé. Dans le cas d'une consultation en tant qu'ambulant, dès que ELEM-DOS est clôturé, il peut être archivé (un flag indique qu'il s'agit d'une consultation non liée à une hospitalisation).

Ici surgit le problème de pouvoir relier les examens techniques, les demandes de laboratoire et aussi les consultations (donc ELEM-DOS) à un élément de dossier (et parfois à une hospitalisation).
Dans les spécifications, il est dit qu'un patient hospitalisé peut se rendre à une consultation où il avait rendez-vous avant son hospitalisation (ou bien qu'il désirait se rendre indépendamment du motif de son hospitalisation en quelque sorte, "profiter" d'être hospitalisé pour aller chez le dentiste...). Un examen technique toujours prescrit par un médecin, peut être demandé par le médecin consulté. Par exemple, un rhumatologue envoye un patient en consultation chez un pneumologue qui demande une radio du poumon. Il est important donc de préciser le lien entre un examen et un élément du dossier. Dans le cas d'un hospitalisé/ou d'un ambulant qui se rend à un examen, il faudra toujours préciser la référence de l'élément du dossier qui en dépend. Il se peut que l'on ait un arbre d'élément de dossier (rarement). Celui-ci permettra de retrouver l'élément de dossier "racine".
Celui-ci n'aura pas de référence d'élement dossier comme attribut. Un examen technique ou de laboratoire aura toujours une référence élément de dossier (ou une référence médecin extérieur).

Une fois un élément de dossier clôturé, une fonction recherche tous les examens techniques, examens de laboratoire ou consultations, qui ont été réalisés pour ce patient pendant son hospitalisation les relie à cet élément de dossier.
Un fois ce travail fait, toutes les parties du dossier sont ordonnées par date d'examen, regroupées par hospitalisation (parfois il faut adjoindre plusieurs éléments de dossier pour une hospitalisation). Les examens techniques et autres non reliés à une hospitalisation, ou à un élément de dossier, sont archivés après avoir été tarifiés (une fois par mois, par exemple).
Les dossiers archivés sont mis off-line suivant les ressources du système.
Cette fonction est uniquement une réorganisation des données. Elle n'est pas nécessairement simple pour cela.

Rem 2 : L'accès aux données médicales archivées ou non

(Function imposed)

Un patient hospitalisé est inscrit à l'accueil. Un lit dans un certain service lui est attribué ainsi que le médecin responsable. Aussitôt, toute personne de l'équipe médicale peut accéder à ces données. Si le dossier est archivé, il est réactivé après demande du secrétariat du service concerné. Dans cette optique, c'est le service administratif qui détermine les accès au dossier médical sans lui-même y avoir accès. Une indépendance des "pouvoirs" est respectée. Une option complémentaire pourrait donner accès à un dossier médical d'un patient qui n'est ni en hospitalisation ni en consultation à condition que le médecin ait soigné le patient (et en imposant un certain délai).

3. Modification des processus en vue du CODAGE

a) Transformation pour le codage

Avant de coder, certaines transformations doivent être faites.

Utilisation d'un pointeur de texte explicite

Celui-ci permet de suivre quelles actions sont possibles ou non pour tel type d'entité. C'est une variable du système qui sera associée à chaque vecteur d'état, bien souvent en entier. Ceci remplace l'utilisation de tests sur les valeurs de records lui-même.

Read et write

Chaque "write" correspond à un appel d'une fonction. Suivant l'état du text-pointeur de l'occurence de l'entité qui effectue une action, celle-ci sera réalisée.
Chaque "read" correspond à un point d'entrée dans le module. En fait, les actions comprises entre deux "read" équivalent à une fonction et sont d'ailleurs suivies immédiatement d'un "return". Le retour de l'appel redonne la main au processus appelant juste après le "write".

Création et destruction du processus

La méthode JSD ne prévoit pas la création des occurences (sauf pour les entités marsupiales). De même la destruction de processus ne concerne pas la perte de spécification de JSD. Ces deux problèmes sont "Out of Model Boundaries". Normalement tous les processus existent dans le modèle dès le départ. La destruction d'occurences se fait uniquement pour une question de ressources. Le principe de JSD est qu'un processus est infini (donc aussi l'occurence). Même si aucune action n'intervient plus, il est toujours possible que l'on ait besoin d'interroger le vecteur d'état de celui-ci. Ces deux décisions ressortent uniquement à la phase d'implémentation.

Appel d'occurence

Chaque fois qu'une action s'effectue sur une occurence, il faut d'abord l'appeler. C'est une instruction "loadsv" qui au début de chaque entrée dans le module (où se trouve la fonction appelée) charge la bonne entité. De même, après que la fonction a traité l'information, il faut stocker l'occurence de l'entité pour mettre à jour la base de données. Une autre structure accomplit cette opération "storsv".
A noter que s'il existe un processus de contrôle au niveau de l'entité comme vu en 4, celui-ci examine le vecteur d'état (text-pointer) pour valider l'action à transmettre. De ce fait, il n'est plus nécessaire de "loader" le vecteur d'état; il peut être transmis comme argument avec l'action et ses attributs.

Le travail du programmeur

La question est : que reste-t-il à faire quand le développement est fait. Dans ce cas-ci, le projet implémenté se limite à automatiser le laboratoire d'analyses et son implantation dans un hôpital. La solution devait tenir compte d'une extension pour l'informatisation du dossier médical.
Ce que le programmeur devra faire.
On peut énumérer les différents points :

- Traduire les processus en langage (cobol, etc...)
- Développer la base de données par processeur "logique"
- Programmer quelques fonctions comme une fonction de recherche (sur base de critères définis - clé partielle, ...) et d'affichage d'une liste d'entités pour faire un choix (médecin, patient,...), ...
- Développer les écrans d'interface, les rapports et les listes
- Programmer quelques fonctions mineures comme celles précisées dans les rubriques fonctions à développer

D'autres fonctions seront nécessaires. Elles pourront avec facilité être développées. Les fonctions "imposed" sont toujours assez simples (accès à la BD) à rajouter. Diverses statistiques pourraient être rajoutées sans beaucoup de transformations.

Processus démembrés

1. DEM-A-PROVISOIRE

a) Pour le laboratoire

```
DEM-A-PROV-2a seq
            *
            *
     read FDAP
            DEM-A-BODY sel (DA-PRO-ACC)
                  DA-PRO-ACC
            DEM-A-BODY alt (DA-PRO-REF)
                  DA-PRO-REF
            *
            *
            *
            *
            DEM-A-BODY end
DEM-A-PROV-2a end
```

b) Pour le service des soins

```
DEM-A-PROV-2b seq
     read AAP
     DA-PRO-INTRO
     read AAP
            *
            *
            *
            DEM-A-BODY alt (DA-PRO-ANN)
            *
            *
            *
                  DA-PRO-ANN
            DEM-A-BODY end
DEM-A-PROV-2b end
```

2. DEM-L-PROV

a) Pour le laboratoire

```
DEM-L-PROV-2a seq
                    *
                    *
     INTRO-ANAL itr
                    *
              read DLP
     INTRO-ANAL end
     DEM-L-BODY sel (DA-PRO-ACC)
            DL-PRO-ACC
     DEM-L-BODY alt (DL-PRO-REF)
            DL-PRO-REF
                    *
     DEM-L-BODY end
DEM-L-PROV-2b end
```

b) Pour le service de soins

```
DEM-L-PROV-2b seq
     read LLP
     DL-PRO-INTRO
     read LLP
     INTRO-ANAL itr
            DA-PRO-INTRO; write DA-PRO-INTRO to LAP
            read LLP
     INTRO-ANAL end
                    *
                    *
     DEM-L-BODY alt (DL-PRO-ANN)
            DL-PRO-ANN
     DEM-L-BODY end
DEM-L-PROV-2b end
```

3. DL-EX-2

a) Pour le laboratoire

```
DL-EX-2 seq
     read DL2&FPL&FVL
     DL-INTRODUIRE
     read DL2&FPL&FVL
     DL-BODY itr
            DL-ACTION sel (DA-INTRO)
            DA-INTRODUIRE; write DA-INTRODUIRE to LA
            read DL2&FPL&FVL
            DL-ACTION alt (DL-MDF)
                  DL-MODIFIER; write DL-MODIFIER to LA
                  read DL2&FPL&FVL
            DL-ACTION alt (DL-ST-MDF)
                  DL-STATUS-MODIFIER
                  read DL2&FPL&FVL
            DL-ACTION end
     DL-BODY end
     DL-TERMINAISON sel (DL-SUPPRIMER)
            DL-SUPPRIMER
                  *
                  *

     DL-TERMINAISON end
DL-EX-2a end
```

b) pour l'administration

```
DL-EX-2b seq
     read FCL
                  *
                  *
                  *
                  *
                  *
     DL-TERMINAISON alt (DL-COMPTAB)
            DL-COMPTABILISER
     DL-TERMINAISON end
DL-EX-2b end
```

4. PATIENT

a) Pour le laboratoire

```
PATIENT-1a seq
                *
     read LP
     PATIENT-BODY itr
                *
                *
                *
                *
                *
                *
                *
                *
                *
          PATIENT-ACTION alt (DL-INTRODUIRE)
                DL-INTRODUIRE, write DL-INTRODUIRE to PDL
                read LP
                *
                *
          PATIENT-ACTION end
     PATIENT-BODY end
PATIENT-1a end
```

b) Pour l'administration

```
PATIENT-1b seq
     read AP
     ENREGISTRER
     read AP
     PATIENT-BODY itr
          PATIENT-ACTION sel (ATTRIBUER)
                ATTRIBUER, write ATTRIBUER to PH
                read AP
          PATIENT-ACTION alt (LIBERER)
                LIBERER, write LIBERER to PH
                read AP
                *
                *
                *
                *
          PATIENT-ACTION alt (SIGN-MDF)
                SIGN-MODIFY
                read AP
                *
                *
          PATIENT-ACTION alt (TRANSFERER)
                TRANSFERER, write TRANSFERER to PH
                read AP
          PATIENT-ACTION end
     PATIENT-BODY end
PATIENT-1b end
```

CHAPITRE III : CRITIQUE ET AVENIR DE LA METHODE JSD

1. CRITIQUE DE LA METHODE JSD

Méthode séquentiellle

La méthode se différencie nettement des méthodes structurées par le fait qu'elle ne se veut pas hiérarchique, à l'exception de l'étape 2 où un type d'entité est décomposé en une suite d'actions (avec certaines contraintes d'ordre) mais les actions sont d'abord décrites dans l'étape 1.
A la fin de l'annexe 1 qui décrit la méthode plus en détail, les arguments de Jackson qui explique bien pourquoi il estime que les méthodes structurées comme Top-down ne sont pas propices au développement d'un système.

Couplage et cohésion

Il me semble intéressant de vouloir utiliser deux critères importants du "design" développés sous les méthodes structurées, auxquels les modules et leurs relations doivent satisfaire pour évaluer JSD.

Le premier critère est le couplage. Il mesure en quelque sorte le degré d'indépendance des modules. Trois facteurs affectent le couplage :
- le nombre de données passé entre modules
  (au plus le nombre de données communiquées est grand, au plus fort est le couplage)
- la quantité de données associée à un contrôle du programme, passée entre modules
  (au plus le nombre de données de "contrôle" est grand, au plus le couplage est fort).
- le nombre de données globales partagées par les modules
  (au plus il y a de données globales partagées, au plus le couplage est fort)

Généralement, on considère qu'il y a cinq types de couplage entre deux modules :

Du plus fort au plus faible couplage :

1. "CONTENT COUPLING" :
Deux modules sont "content coupled" si un module réfère ou change les valeurs internes d'un autre module.

2. "COMMON COUPLING" :
Deux modules sont "common coupled" s'ils partagent les mêmes valeurs globales.

3. CONTROL COUPLING :
Deux modules sont "control coupled" si les données de l'un sont utilisées pour diriger les ordres des instructions dans les autres.

4. "STAMP COUPLING" :
Deux modules sont "stamp coupled" s'ils communiquent à travers une structure de données. Les données contenues dans la structure ne sont pas toutes utilisées bien que celle-ci soit transmise entièrement à un autre module.

5. "DATA COUPLING" :
Les deux modules sont "data coupled" s'ils communiquent à travers une variable ou un tableau qui est passé directement comme un paramètre entre deux modules. Les données sont utilisées pour un traitement orienté vers le problème concerné et non à des fins de contrôle du programme.

Le second critère est la cohésion. Autant le couplage mesure le degré de relation entre modules, autant la cohésion mesure la relation entre les éléments des modules. Sept niveaux de cohésion sont classiquement décrits (du plus fort au plus faible) :

1. "FUNCTIONNAL" :
Chaque élément dans le module est une part essentielle d'une et de seulement une fonction.

2. "SEQUENTIAL" :
Les éléments d'un module sont reliés en appelant différentes parties d'une séquence d'opération où la sortie d'un traitement est l'entrée du suivant.

3. "COMMUNICATIONAL" :
Les éléments d'un module opèrent tous sur les mêmes données.

4. "PROCEDURAL" :
Les éléments d'un module font tous partie d'une procédure (une certaine séquence d'étapes qui sont exécutées dans un certain ordre.

5. "TEMPORAL" :
Les éléments d'un module sont reliés au facteur temps sans qu'il y ait un certain ordre ou qu'ils travaillent sur les mêmes données.

6. "LOGICAL" :
Les éléments d'un module sont tous orientés pour effectuer une même classe de traitement.

7. "COINCIDENTAL" :
Les éléments d'un module sont essentiellement sans aucune relation tant au niveau des fonctions que des procédures, données ou quoi que ce soit d'autre.

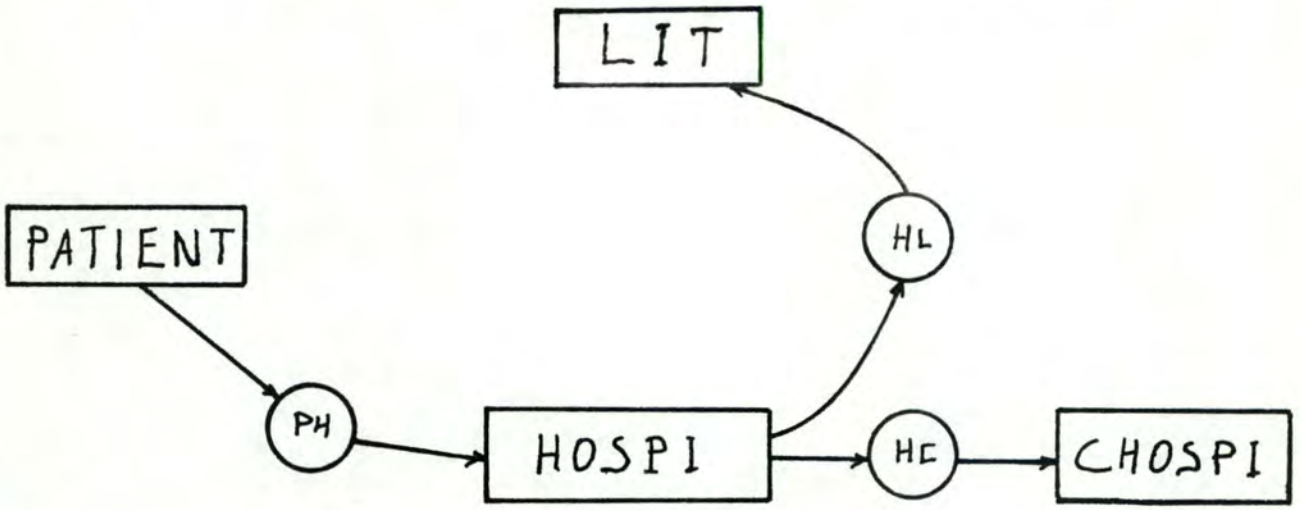
Ces deux critères, dans les méthodes structurées, déterminent aussi le degré de facilité avec laquelle un programme peut être modifié.
On souhaite un programme avec des modules qui possèdent une forte cohésion et un faible couplage entre eux.

Dans JSD, un module correspond en fait à un processus d'un type d'entité (DL-EX, DP-EX, DA-EX,...). C'est un module logique. Il possède un certain nombre de points d'entrée associé à un même processus. On peut dire que les modules sont très faiblement couplés. Il n'y a jamais de données qui soient transmises, qui contrôlent le programme. Tout le contrôle est défini par la séquence d'action et la valeur du "Text-pointer" associé à chaque vecteur d'état. (Les modules de niveau 1 ne sont pas considérés quand il existe un niveau 2).

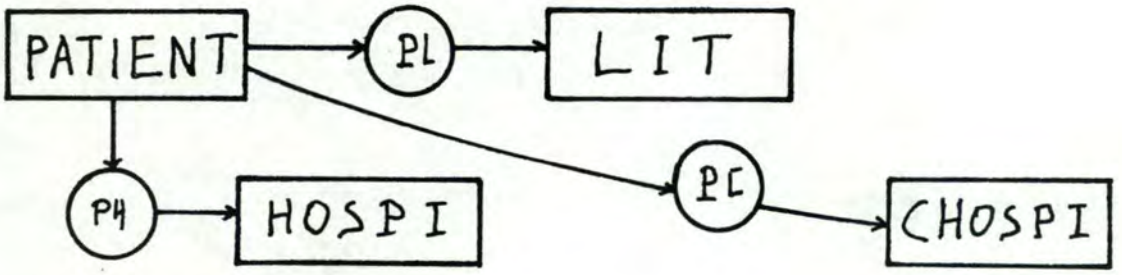
La cohésion est très forte. Chaque module s'occupe d'une seule fonction logique qui correspond au processus d'une entité.
Après passage du SSD au SID, des "supermodules" logiques sont construits.

Dans l'exemple vu, on aura un "super module" LABO, SERVICE DE SOINS, SERVICES TECHNIQUES, ADMINISTRATION et MEDECIN. Le couplage entre ces modules est très faible et la cohésion dans chacun d'eux très forte. A ce point de vue, la méthode JSD apporte un maximum de garantie.

On pourrait objecter que dans certains cas le couplage n'est pas maximal.



Dans le cas ci-dessus, il se peut que des données transmises à HOSPITALISATION sont destinées à LIT ou à CELEM-DOS. Le couplage devient de type "STAMP". Si l'on désire conserver à tout prix un couplage maximal, par exemple pour un programme qui a de fortes chances d'être modifié, une nouvelle construction du SSD peut se faire (voir le paragraphe en II.3.3.).



Spécification et implémentation

Critère essentiel des méthodes structurées, la phase d'implémentation est radicalement séparée de la phase de spécification. Ce qui fait que l'on peut prendre des décisions différentes au niveau de la réalisation sans pour cela que changent les spécifications. L'impact sur le schéma de l'architecture physique préconisée tout comme la possibilité de construire un propre "scheduler" donne une appréhension beaucoup plus grande lors du développement.

Base de données

Une critique est émise par Mr. Hainaut dans son livre (Conception de la base de données) en introduction page 1-2 à ce point de vue :

"Pris au sens moderne du terme, le concept de base de données se voit associer deux objectifs. Le premier est d'être une représentation fidèle d'un système réel (ce que l'on appelle aujourd'hui le "réel perçu"), essentiellement dans ses aspects statiques. Le second est de constituer un serveur de données opérationnel et efficace pour une gamme de traitements."
.....
"Or on constate chez de nombreux auteurs que la phase de mise-en-oeuvre et parfois même la construction entière de la base de données sont réduites à leur plus simple expression, voire même ignorées dans certaines démarches de développement de Systèmes d'Information. Certaines approches de conception dites "par les traitements" ou du type "software engineering" ne conçoivent encore souvent les données externes que comme de simples sources dont le contenu et la structure doivent être adaptés aux besoins des programmes. La définition des fichiers découle de l'analyse des traitements dont elle n'est qu'un sous-produit secondaire. On ignore ainsi complètement le premier rôle d'une base de données (représentation fidèle d'un système réel), pour ne retenir que le second (serveur de données efficace). Les défauts de cette approche sont bien connus puisqu'ils servent de base à la comparaison traditionnelle des approches "fichiers" et "bases de données". Ils se traduisent généralement par un risque important d'instabilité de la définition des structures de données vis-à-vis de nouveaux besoins. On trouvera par exemple dans Jackson, 83 un paragraphe de deux pages consacré à la conception de la base de données, et qui consiste principalement à en démontrer l'inutilité. Sur ce point, l'approche de Jackson relève de celle des types abstraits comme moyen d'expression de schémas externes (voir Chapître IV du présent volume). Le problème de leur synthèse en vue de constituer un schéma global, stable et général (c'est-à-dire conceptuel) n'est pas abordé."

Dans le premier paragraphe, il est stipulé qu'une base de données se doit d'être une représentation fidèle d'un système réel, essentiellement dans ses aspects statiques. Tout le problème est de savoir si une réalité dynamique peut être "modélisée" de prime abord par une réalité statique (base de données). Jackson répond non. Pour lui, c'est le système lui-même qui doit contenir en premier lieu la réalité dynamique. Cette réalité est ramenée à une réalité statique à travers les spécifications.C'est le premier rôle d'un système de représenter fidèlement le monde réel. Le système s'appuie sur une base de données. La réalité statique est secondaire (dans le sens de seconde), superdonnée du système.
La question qu'il faut peut-être poser : est-que le schéma entité-association peut-il représenter la part essentiellement statique d'un système, en tant qu'image du monde réel. (Jackson répondait non, vraisemblablement).
La méthode de conception d'un système développé par Bodart et Pigneur utilise notament un schéma entité/association et un diagramme de la dynamique des flux.

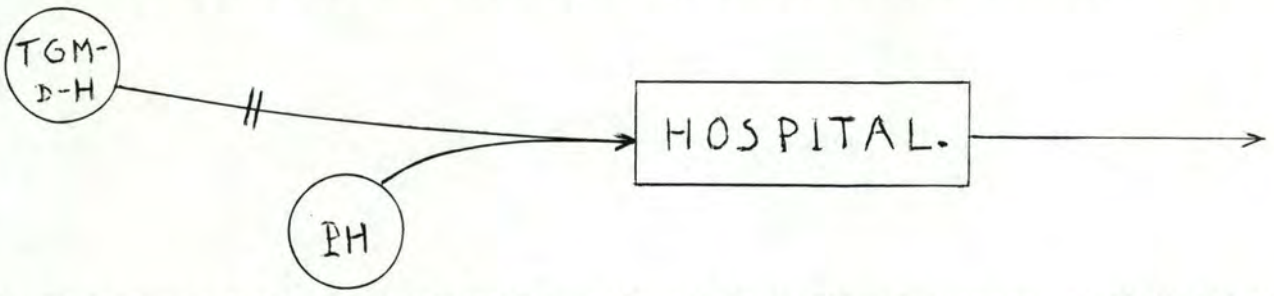
Dans JSD, une suite d'étapes permet de traduire principalement à l'aide de trois outils diagrammatiques interconnectés (diagramme de structure des types d'entité - + texte en pseudo-langage -, SSD et SID) un système. Les spécifications sont établies progressivement au travers des différentes étapes et finalement il se fait que la spécification devient, tend à être le programme (en pseudo-langage ou directement en langage de programmation).
La démarche de la conception du système avec JSD est prioritaire par rapport à celle de la conception de la base de données. Celle-ci ne peut se faire en parallèle, indépendamment. Cependant, toutes les étapes de transformation et d'évaluation des algorithmes restent applicables bien que les transformations se fassent plutôt à l'envers. Cette méthode se prête aisément aux bases de données distribuées conjointement à des processus distribués.

Outils diagrammatiques

La part de l'outil diagrammatique est essentiel. Ils ne sont pas toujours facile à gérer. La complexité d'un système (nombre de type d'entité et nombre d'action par entité très élevé) arrive vite à des dessins complexes. Dans l'essai de développement de ce mémoire, la réalisation et la présentation de ceux-ci ont été particulièrement difficiles et probablement critiquables. Il me semble que le développement d'une série d'outils graphiques devrait permettre de mieux "manager" cette complexité et de pallier à cet écueil.

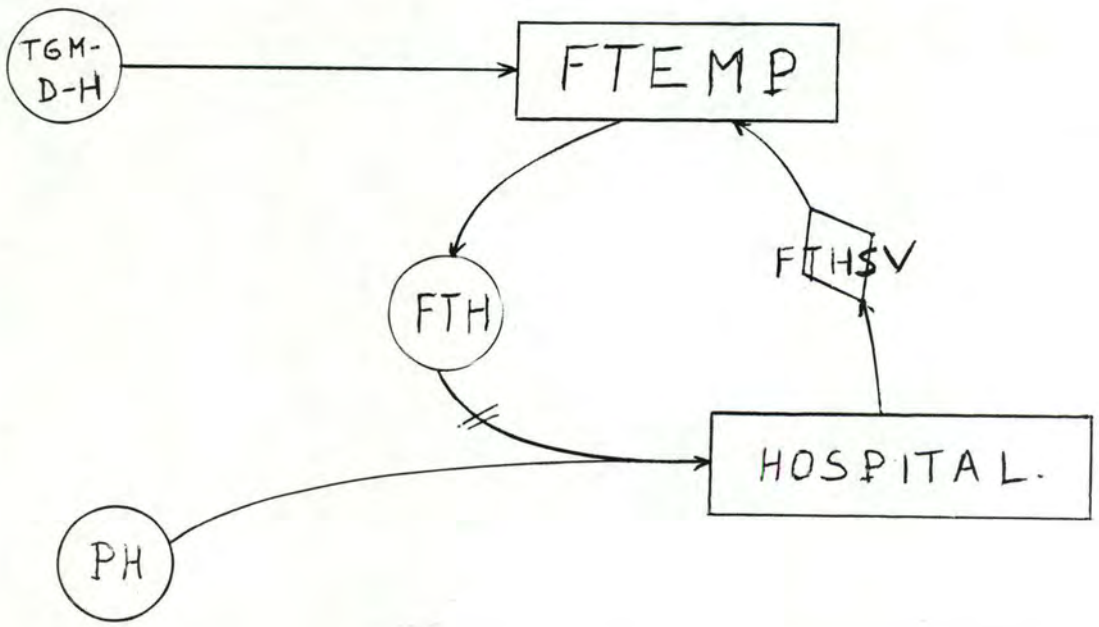
Traduction incomplète de certains processus

Un second point qui devrait être approfondi se trouve dans la mauvaise représentation d'une distribution d'une donnée, d'un processus vers plusieurs autres.

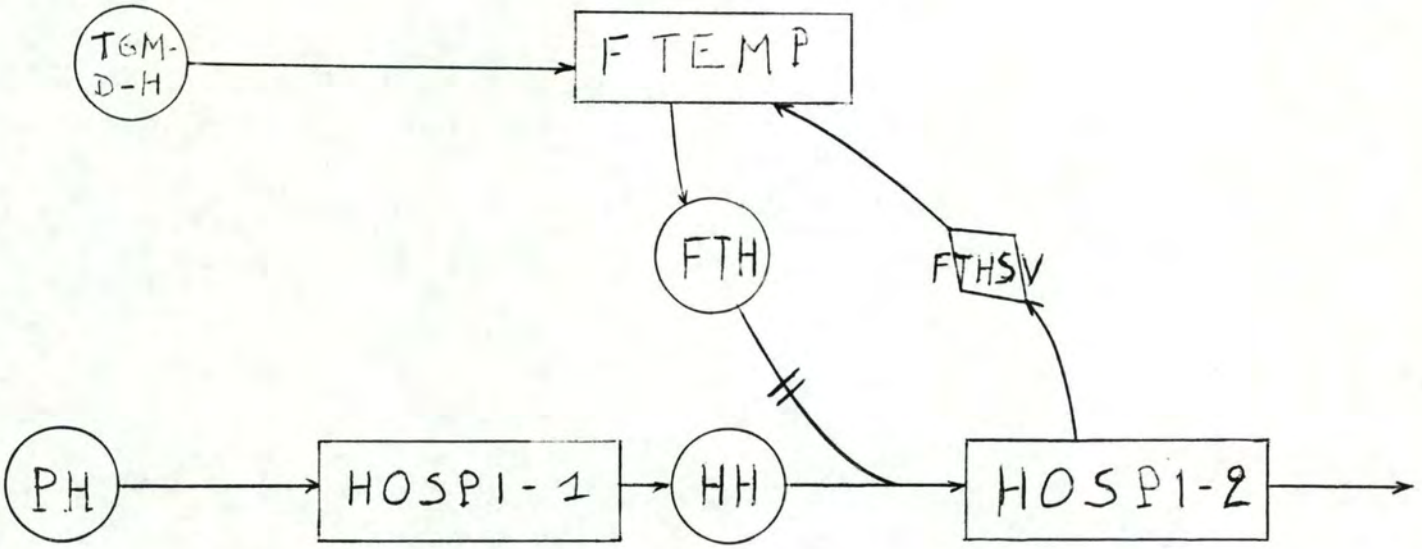


Un message du système d'exploitation indique à toutes les occurences d'HOSPITALISATION qui se trouve après une attribution et avant une libération, que telle action (un évènement du monde réel) vient de se passer. Cette information est signifiée comme étant un "data-stream". Cependant à la phase d'implémentation et construction du SID, c'est-à-dire où l'on s'adresse à une foule de processus avec un seul texte, il faudra atteindre ces différentes entités et on devrait avoir plutôt un schéma SSD de ce type :

FTEMP serait une fonction interactive et nécessiterait un level-2.

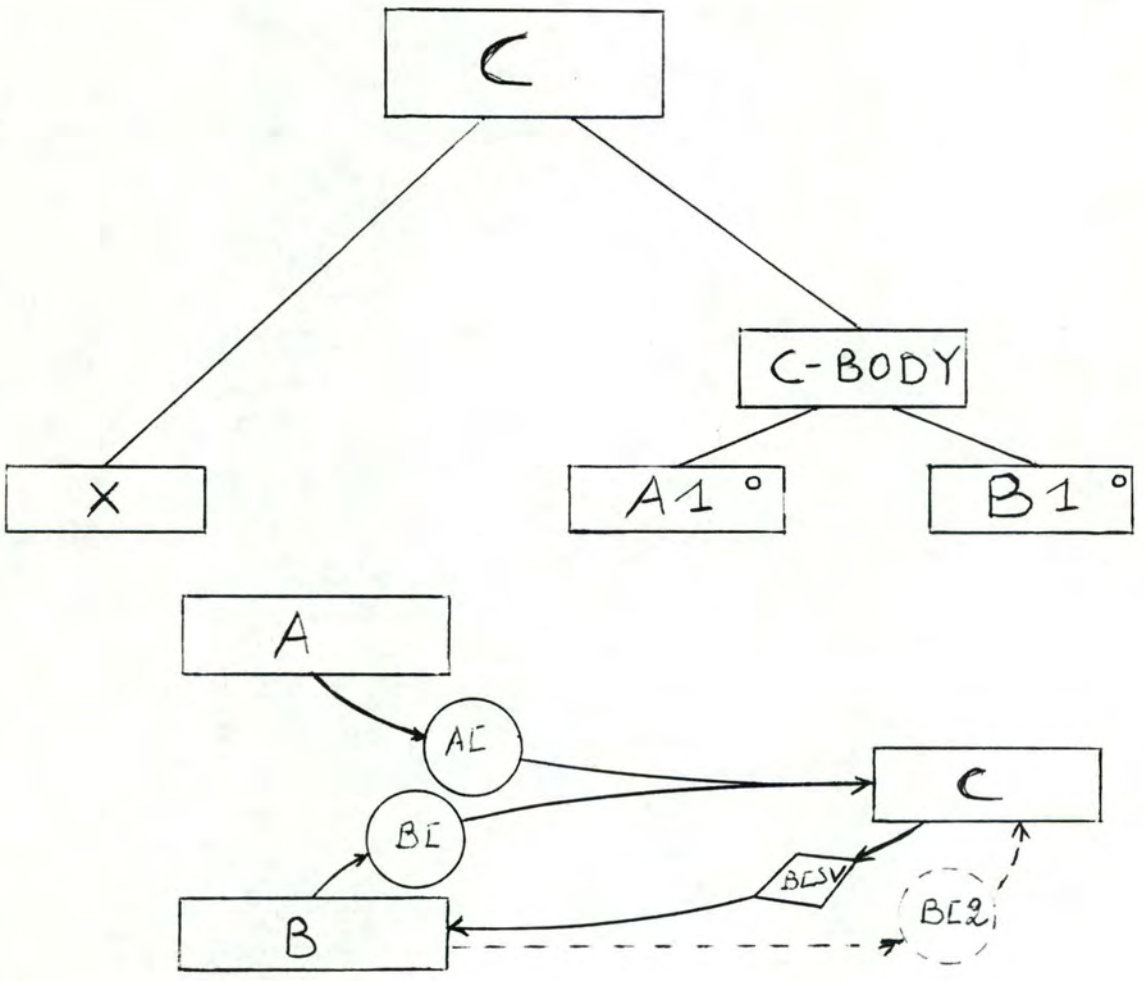


Nouveau SSD :



Un problème similaire se pose quand Jackson propose une manière de gérer la concurrence des actions sur un même processus, actions transmises par des processus différents de façon alternative.

Exemple simplifié :



Les actions A1, A2 proviennent de A, les actions B1, B2 proviennent de B. Le problème surgit premièrement du fait qu'un "rough merge" n'est pas "fair". Quand une action est envoyée à un processus C par A avant une action envoyée à ce nouveau processus (C) par B, la première n'arrive pas nécessairement avant les secondes.

D'autre part, si un processus B via une recherche par un vecteur d'état sur C, envoye par la suite une action (B1), un conflit peut se présenter si dans l'intervalle une autre action (A2) venant de A est envoyée. Etant donné l'avancement du text-pointer de l'entité, il n'est plus possible de lire l'action B1.

JSD détecte admirablement bien ce genre de conflit : la solution proposée par contre a le même défaut que le problème précédent. Il propose d'envoyer à la place de la liaison "state-vector", un "data-stream" sur l'occurence qui interdit toute action venant d'un autre processus (le seul "read" accepté sera celui de B).
Mais comment reconnaître l'occurence de C quand il s'agit d'une opération interactive? Il me semble qu'il faut donc d'abord un "state-vector" pour identifier l'action.
Ou bien, quand le vecteur d'état est inspecté, un "read lock" est mis sur "lui", ce qui fait que tout accès de C pour un autre processus n'est plus possible (solution moins élégante).
La façon de résoudre ce genre de conflit nécessite quelques analyses (le résultat peut être un nouveau symbole graphique, par exemple).

2. LA PROGRAMMATION "OBJECT-ORIENTED" et JSD

La POO (Programmation Objet-orienté) n'est pas si récente mais elle présente un regain d'intérêt ces derniers temps. Le système d'exploitation MacIntosh d'Apple, application pratique par excellence de ce type de programmation, est bien connu du grand public. Smalltalk-80 est aussi célèbre parmi les informaticiens, Clue un peu moins.

Dans la POO, un objet est d'abord défini par une classe. Une classe peut-être comprise comme un nouveau type pour une collection d'objets. Un objet contient à la fois des variables locales et un ensemble de méthodes (ou fonctions) associées pour manipuler ces données. Une fois un objet déclaré, il faut retenir des variables.

Un concensus s'établit progressivement pour affirmer que les éléments nécessaires pour qu'un langage supporte la POO sont au nombre de quatre.

L'information cachée reprend l'idée classique de la boîte noire. Certaines procédures manipulent des données mais la manière de le faire est cachée.

L'abstraction de données est à considérer comme une façon d'utiliser l'information cachée. Les données ne sont plus définies par leur représentation concrète. Elles deviennent abstraites dans le sens que ce sont par des actions qu'elles subissent qu'elles sont définies.

Le "Dynamic binding" exprime le fait que c'est à l'exécution que la liaison d'un objet et sa méthode (ou fonction) se font automatiquement et non à la compilation (late binding).
Auparavant, il fallait prévoir à la compilation, par exemple, un "case statement" qui en fonction de chaque type appelle une fonction différente (implémentation de la fonction print( ) en C par exemple-early binding).

L'héritage ("inheritance") autorise une classe d'objets à être dérivée d'une ou plusieurs autres (multiple inheritance) classes (appelé parfois superclasse (smalltalk) ou base-class (c++) et les méthodes associées à ces classes "souches" sont héritées sans que la définition de la classe devienne plus complexe (code factoring).

Beaucoup de langages apparaissent sur le marché avec certaines de ces propriétés. Peu cependant les offre toutes (C++, Objectiv-C, Actor,...). D'autres langages comme le Lisp ont rajouté un noyau permettant de faire de la POO (Scoop/PC Scheme, Flavor, Common Loops, ...). La "multiple inheritance" et l'"Automatic Storage Management" (gestion de la mémoire optimale) sont deux écueils que quelques implémentations seulement arrivent à résoudre.

La seconde notion sous-jacente est le concept de message entre objets. Au lieu de passer des données à une procédure, on demande aux objets d'appeler des opérations à exécuter sur eux-mêmes en s'envoyant un message ou à d'autres objets (d'une classe différente). La responsabilité de l'exécution d'une opération est donnée à l'objet lui-même.

Quand on examine la méthode JSD, tout le monde peut constater cette similitude avec les types abstraits. Mr. Hainaut l'indique bien dans l'extrait repris quelques pages plus tôt.

Chaque type d'entité est superposable à une classe d'objet. Cette classe serait définie par les actions du type d'entité concernée. Toute la méthode JSD est basée sur la communication entre processus (informations ou messages). Toutes les particularités de JSD rejoignent celles de la POO. Aussi ce type de programmation sera le plus appropriée. L'avenir de la POO semble prometteuse principalement à cause du factoring. JSD peut certainement profiter de cet essor.

IV. CONCLUSION

Le travail s'achève ici. Il est certes incomplet. Bien des particularités de JSD au niveau notamment des inversions de processus, du "batch processing",... de la réalisation de "channel", des processus de "scheduling", de l'utilité du "back-tracting", n'ont pas été montrés lors de ce développement (cfr Annexe A). D'autre part, le système proposé n'est détaillé que sur une petite partie (la partie laboratoire). Celle-ci n'a pas été poussée dans ses moindres détails. Le programmeur devra passer encore un certain temps avant la réalisation concrète bien que la programmation, au sens large, est cependant répartie dans plusieurs étapes du développement. A cela, il a été préféré de s'attarder sur les points essentiels de cette méthode récente (1983) délaissée, boudée même par les informaticiens. Certes, des améliorations sont nécessaires, surtout au niveau des outils, pour faciliter son emploi. Beaucoup de programmeurs de Cobol qui s'y sont essayés ne l'aiment pas; en effet, le type de programmation (orienté objet) n'est pas l'apanage du bon vieux Cobol.

Pourtant, JSD a beaucoup d'atouts. Il faut la considérer, à mon avis, comme une couche abstraite au-dessus de la décomposition fonctionnelle (et des méthodes structurées). Celle-ci a tout-à-fait sa place (dans le cas de ce travail, plusieurs fonctions mineures y ont recours) mais dans une phase suivante. Les griefs que Jackson leur reproche au niveau méthodologique pour le développement de gros systèmes sont fondés. Le principal étant que les décisions les plus importantes, qui ont le plus de répercussions sur le développement, sont prises au tout début. Erreur que JSD évite admirablement. Quand le développement est fini suivant JSD, la méthode de décomposition fonctionnelle peut très bien prendre le relais pour ce côté plus lié au codage. Elle ne s'applique plus dès lors qu'à des fonctions parfaitement connues et définies (écran, rapport, algorithme de facturation, ...). Elle ne subit plus les désavantages qui lui sont liés (cfr annexe A-Conclusion).

L'idée maîtresse de JSD est de défendre le principe que le modèle est prioritaire sur les fonctions. Parce qu'il n'y a pas cette distinction entre "model process" et "function process" (en précisant que le modèle est plus stable que les fonctions et que le modèle doit être développé dans un premier temps). Les méthodes hiérarchiques de développement (Top-down, Warnierr-Orr,...) n'arrivent pas à garder une appréhension du monde réel aussi forte que JSD tout au long du développement. Aucune méthode n'arrive non plus à un résultat aussi proche de l'affirmation : "la spécification est le programme".

L'avenir de cette méthode peut compter sur une automatisation (avec les avantages de multiples contrôles d'intégrité) aisément réalisable. Le couplage à un générateur de code de type HOS ne demanderait pas énormément d'effort et promet de rendre JSD encore plus passionnant.

L'informatique actuellement doit s'occuper de développer des systèmes d'information complexe et probablement s'orienter vers une distribution de ressources tant au niveau matériel que logiciel (programmes et données). A ce type de besoins, de nouvelles méthodes doivent permettre de répondre. JSD apporte si non pas une solution miracle, au moins une voie très prometteuse.

TABLE DES MATIERES

LITTERATURE CONSULTEE

BODARD, F et PIGNEUR, Y. : Conception assistée des applications informatiques
T1; Etude d'opportunité et analyse conceptuelle,
Masson, 83

COX, J.B. : Object-oriented Programming,
Addison-Wesley, 86

JACKSON, M. : Principles of program design
Academic Press

JACKSON, M. : System Development,
Prentice-Hall, 83

HAINAUT, J-P. : Conception assistée des applications informatiques
T2; Conception de la base de données,
Masson, 86

ROGER, F.M. : Le résumé du dossier médical
Edition privée

STROUSTRUP, B. : The C++ programming language
Addison-Wesley, 86

WEED, L.L. : Medical record that guide and teach
N. Engl. J. Med. 278 : 593-599, 1968

APPENDIX A

SYSTEM DEVELOPMENT de M. JACKSON

Introduction

Michael JACKSON présente une méthode de développement d'un système d'information. Cette méthode (dont l'acronyme est JSD) et ses principes sont développées ci-dessous.

Méthode et principes.

Le principe fondamental de JSD est de construire un modèle du monde réel. Le système est considéré comme une sorte de simulation du monde réel. Le modèle est une description abstraite donc partielle du monde réel. La finalité du modèle (donc du système) déterminera l'inévitable sélectivité (partialité) d'une telle description. Une fois le modèle élaboré, il existe une base conceptuelle pour les fonctions qui seront greffées sur celui-ci. La description abstraite ainsi que sa réalisation forment ensemble le contexte pour la spécification fonctionnelle. Ils définissent un ensemble de mots qui seront utilisés dans la spécification des différentes fonctions. La signification de chaque mot dans le monde réel est donnée par la connection entre le monde réel et le modèle. De ce fait, les fonctions peuvent être définies de façon non ambigüe. Le modèle, de l'application, définit implicitement un ensemble cohérent de fonction. Le modèle est plus stable que les fonctions et cela parce que les fonctions sont exprimées en terme dont la signification est définie par l'angle sous lequel la réalité est perçue par l'utilisateur. Si la réalité est perçue sous un angle différent, les fonctions doivent changer mais l'inverse n'est pas vrai : les fonctions peuvent changer sans pour cela que la vue sous-jacente de la réalité doive se modifier. Cependant il est nécessaire de connaître une certaine idée des différentes fonctions pour déterminer le

modèle. Bien que la méthode ne prévoit l'ajout des fonctions qu'à partir de la quatrième étape, le modèle contient la capacité de supporter l'adjonction des fonctions. Le développement du système s'appuie nécessairement sur la connaissance des fonctions requises. Les applications où principalement les évenements se déroulent dans un certain ordre (c-à-d où la dimension temporelle est de première importance) décrivent des réalités que l'on peut appeler dynamiques. JSD s'adresse particulièrement à cette classe d'application.

Un second principe de cette méthode, en effet, précise qu'une base de données ne sait pas modeliser un "monde réel" dynamique. Une "réalité dynamique" requiert un modèle dynamique (par opposition à une réalité statique qui peut être reproduite par une base de données).
Un modèle dynamique est composé d'un certain nombre de processus séquentiels. La forme d'un processus séquentiel est celle d'un simple programme structuré (pas de paralléllisme, ni de concurence ou de multitâche à l'intérieur d'un simple processus séquentiel). L'addition d'une fonction (et du "output system") se fait une fois le modèle construit. La fonction est directement incorporé dans le "model process" dans les cas simple ("embedded function") mais bien souvent il faut introduire un nouveau processus, différent des processus du modèle ("imposed ou interactive function"). Les processus sont connectés soit entre eux, soit avec le monde réel (input,output). Il existe deux types distincts de connection : soit par un message séquentiel ("data-stream connector"), soit par l'ensemble de toutes les variables d'un processus ("state-vector connector").

Dans les deux premières étapes de la méthode JSD, une description abstraite est faite qui permet de spécifier un modèle de la réalité (sous la vue voulue) dans la troisième étape. Le monde réel sera décrit en terme d'entité, d'action qu'elles produisent ou subissent, et de l'ordre dans lequel celles-ci s'exécutent. La notion d'entité est intimement associée à l'idée de processus du modèle et à l'idée d'un ensemble ordonné d'action. L'adjonction des fonctions est réalisée au cours de l'étape suivante. Une étape de spécification au niveau des temps de réponse peut prendre plus ou moins d'importance suivant l'application (temps réel,...).

La deuxième partie s'occupe de l'implémentation. Les processus dans JSD sont toujours exécutés une seule fois et leur exécution ont la même durée de vie que l'entité elle-même. Ces processus sont en principe directement exécutable en utilisant un processeur pour chacun de ceux-ci. En réalité, ceci n'est pas concevable et il faut passer par un OS sur une seule machine avec gestion des tâches. Cependant, la manière d'implémenter ces différents processus peut être déterminé quand le système est construit plutôt que quand il s'exécute sur un OS dont la gestion multi-tâche n'est pas connue. Des choix divergents à ce niveau vont donner des implémentations différentes. Certaines transformations sont requises pour passer d'une machine multi-processeur théorique à une machine mono-processeur réelle. L'inversion de programme, par exemple, permet la suspension et l'activation d'un processus. On peut implémenter de cette manière la gestion de plusieurs processus. L'utilisation d'OS prévu pour la programmation concurrente est une autre solution.

Procédure de développement.

Comme esquissée plus haut, la procédure de développement se fait en six étapes.
Les cinq premières étapes concernent la création des spécifications du système à développer :

2 étapes - description abstraite du modèle
1 étape - réalisation du modèle
1 étape - adjonction des fonctions
1 étape - spécification des contraintes d'ordre temporel

La deuxième partie, constituée d'une étape, correspond à l'implémentation :

1 étape - implémentation

1 ENTITY ACTION LIST

Résumé : la première étape définit l'application en décrivant "le monde réel" en terme d'entité et d'actions qu'elles appelent ou subissent. Il en résulte une liste d'entité et une liste d'action qui concerne une ou plusieurs d'entre elles. De plus, pour chaque action, est associé une description informationelle de l'évenement du monde réel. Les entités sont décrites uniquement par l'ensemble des actions qu'elles supportent. Une entité de JSD est complètement décrite, à ce stade de développement par ses actions. Il en résulte une description hautement abstraite du "monde réel". Le système doit, éventuellement, être capable de fournir des rapports qui peuvent être exprimés dans les termes de cette description. Les listes (des entités et des actions) forment une sorte de glossaire des termes qui peuvent plus tard être utilisés pour spécifier les fonctions du sytème; elles délimitent les différentes fonctions possible du système en définissant les mots par lesquels elles peuvent être exprimées.

Il faut mettre en évidence les entités et les actions s'y rapportant.
Une entité doit satisfaire à ces 3 conditions dans JSD :

- appeler ou subir des actions dans un ordre déterminé.
- exister dans le monde réel et non dans le système lui-même.
- être capable d'être regardé comme individuel tout en étant nommé univoquement, s'il y a plus d'une entité d'un type.

Une action doit satisfaire à ces trois conditions dans JSD :

- elle est regardée comme prenant place à un point particulier dans le temps, plutôt que persistant sur un laps de temps.
- elle prend place dans le monde réel à l'extérieur du système et n'est en aucun cas une action du système lui-même.
- elle est regardée comme atomique et n'est pas décomposable en subactions.

La définition des entités et de leur actions correspond aux fondations du modèle. Il est interessant de citer les modifications principales apportées à ces listes :

- le rejet d'actions dans la seconde étape (ENTITY STRUCTURE STEP) parce qu'elles apparaissent appartenir au système plutôt qu'au monde réel,
- le rejet d'actions dans la troisième étape (INITIAL MODEL STEP) parce qu'elles ne peuvent pas être facilement et économiquement détectées et signalées au modèle,
- l'ajout d'une entité dans la deuxième étape (ENTITY STRUCTURE STEP)

parce qu'elle est nécessaire pour spécifier l'ordre des actions d'une entité existante,
- l'ajout d'une action dans la troisième étape (INITIAL MODEL STEP) parce qu'elle est nécessaire pour l'interprétation d'actions existantes.

Le choix des entités est limité par deux considérations :
1. Si certaines actions sont possible sur seulement une partie des instances d'un type d'entité, il est nécessaire de créer un second type d'entité.
2. Il faut spécifier la vie entière d'une entité. Il n'existe pas d'occurence d'entité qui change de type pendant sa durée de vie.

Remarques
- Une entité peut être collective si ses actions se font en groupe.
- Si un type d'entité semble justifié et les fonctions qui sont basées sur celle-ci souhaitées, le développeur doit tenir compte du prix de l'inclusion de ce type d'entité dans la description abstraite. On peut estimer raisonnablement que le coût dépend de deux facteurs :
  - le coût du développement et de la maintenance du système est ± proportionnel à l'inverse du nombre de type d'entité déja inclus,
  - le coût de l'exécution qui dépend du nombre d'occurence de ce type d'entité et du nombre d'action de chaque occurence.
- Une action commune est une action qui fait intervenir plusieurs entités (parfois de même type). C'est une décision de modélisation.

2 ENTITY STRUCTURE STEP

Résume : les actions de chaque entité sont ordonnées dans le temps. Il apparait nécessairement différentes contraintes d'ordre, qui sont exprimées ici par un diagramme. Celui-ci constitue une spécificaton plus exact du comportement d'une entité que la liste d'action seule. Cette approche à l'avantage de la précision et surtout de l'instantanéité de l'appréhension. Un diagramme est construit par entité (au minimum).

Le but de cet étape est d'ordonner les actions qui concernent les entités respectives, en precisant certaines contraintes si nécessaire. La representation utilisée est identique à celle que Jackson utilisait dans une méthode connue sous l'acronyme JSP (cfr bibliographie). Il faut espérer qu'un seul diagramme suffisse pour chaque type d'entité. A ce momment , il doit seulement exprimer les structures minimales pour simuler les contraintes du monde réel. Parfois plus d'un diagramme est nécessaire pour spécifier les contraintes temporelles et l'addition d'un diagramme suggère de nouveaux types d'entité qui n'apparaissent pas dans la liste originelle. Le diagramme doit couvrir toute la durée de vie de ces entités et non seulement une partie de ceux-ci.

Une entité marsupiale est une entité qui émerge de la structure d'une autre entité, spécialement quand plusieurs instances de l'entité marsupiale existent concourement dans une instance de l'autre. En effet, un simple diagramme ne peut exprimer la concurrence. Si une entité peut exécuter différentes actions, et concourement sur des "entités cachés" avec des contraintes d'ordre sur chacune de celles-ci, le comportement de cet entité ne peut pas être montré en un seul diagramme et il est nécessaire d'extraire le type d'entité masqué et de lui associer un diagramme propre. Ce type d'entité est appelé marsupiale par analogie.
Parfois la concurrence apparaît quand une entité joue plusieurs rôles et que l'ordre des contraintes de rôles ne peuvent pas être combiné en une seule structure. Il est nécessaire de spécifier un diagramme séparé pour chaque rôle. La terminaison prématurée de la vie d'une entité est difficile à exprimer dans un diagramme. Une sélection spéciale est nécessaire.
Une entité peut aussi être une composition de deux ou plusieurs entités.

3 INITIAL MODEL STEP

Résumé : le résultat des deux premières étapes est une description abstraite du monde réel en terme de processus séquentiels. Dans cette étape, le développeur commence à spécifier le système lui-même en spécifiant une simulation du monde réel. La tâche est d'établir comment le "real world process" est connecté au processus du modèle (le mode de connection standard est le "data-stream connector"). Pour chaque action du monde réel, un message est entré dans le processus du modèle dans le système. Un nouveau type de diagramme est utilisé et est appelé System Specification Diagram (SSD). Le développeur spécifie le processus du modèle en écrivant un texte structuré. Ce texte en pseudo-code est essentiellement une forme textuelle du diagramme du processus mais en incluant les opérations d'entrée de communication.

A ce niveau, il faut établir des connections entre processus et le monde exterieur. Les connections possible sont au nombre de deux.
Le "data-stream" : un processus écrit une suite de donnée (consistant en un ensemble ordonné de message et/ou de record) qui est lue par un autre processus (read, write).
Le "state-vector" : un processus inspecte directement les variables d'un autre processus (uniquement possible quand il existe une lecture ou une écriture d'une suite de donnée) (getsv).

Pour spécifier complètement un processus séquentiel, on ajoute une notation textuelle (pseudo-code JSP) au diagramme. La base de cette notation est simplement une transcription de la forme diagrammatique (seq - itr - sel). Une nouvelle représentation diagrammatique est utilisée : le SSD. Elle incorpore les data-stream et state-vector.

Dans la représentation des connections entre les différents processus, elle traduit les informations relatives au type de relation entre entité (1-N, N-1, N-N). Dans la notation textuelle, les opérations de connections sont introduites à l'endroit ad hoc.
Remarques.
- Le monde réel est représenté par le niveau 0.
- L'allocation des opérations de connections est systématique et impérative (pas de poursuite du processus si l'allocation n'a pas été faite).
- Il est interessant de relever :
  pour le data-stream : quand les données transmises ne contiennent pas uniquement un record pour chaque action dans le processus emetteur, il est quelques fois utile de décrire la structure des données transmises, séparement de la structure des processus connectées eux-même.
  pour le state-vector : paralléllement pour les vecteurs d'état, il peut être utile de décrire leur succession comme s'ils étaient une suite de record et de décrire leur structure explicitement.
- Quand une suite de donnée connecte deux processus, elle contient une description abstraite des deux; le fait qu'elle peut donc incorporer une description des deux processus montre clairement que leur comportement doit être coordonné pour satisfaire la description de data-stream. Ceci reste vrai pour plusieurs processus emetteurs (rough merge).

Il arrive que plusieurs "input" convergent vers un processus :
- "fixed merge" : les messages sont mélangés en fonction d'une règle préétablie,
- "data merge" : les messages venant des processus sont mélangés en fonction de leur contenu (ceci implique une lecture en avant et peut donc bloquer un processus très longtemps),
- "rough merge" : certains facteurs rendent indéterminé la façon dont les messages sont mélangés (il n'est jamais équitable).

Dans les processus du modèle spécifié, le passage du temps dans le monde réel est marqué par l'occurence d'action, signalée au modèle par un record dans le flux d'entrée. Parfois, cependant une action doit être déclenchée dans une période de temps bien déterminée, par exemple. Pour signaler au modèle ce genre particulier d'évenement, une entrée appropriée au modèle est réalisée : ce sont les TGM (Time Grain Marker).

Création et destruction de processus
Aucun processus n'est créé durant la durée de vie du système et aucun n'est détruit. Pour chaque type d'entité, le développeur doit décider où placer la responsabilité pour la création de nouvelles entités (de ce type). La création de nouvelles entités (donc processus) se trouve en dehors du modèle pour la raison qu'on ne sait pas attribuer à un certain type d'entité la responsabilité de la création. Celle-ci regarde plutôt l'implémentation que la spécification. Seule les

entités marsupiales peuvent être créées par leur entités parentales. La destruction d'un processus concerne toujours la phase d'implémentation dans JSD. Un processus ne peut en aucun cas se détruire lui-même ni en détruire d'autres. Le but implicite est toujours de libérer des ressources occupées par des processus.

4 FUNCTION STEP

Résumé : les trois premières étapes ont produit la base sur laquelle les fonctions du système sont à greffer. Dans cette étape, ces fonctions sont spécifiées en utilisant les termes du modèle. Une combinaison d'évenement du monde réel est modelisée par une combinaison d'évenements du système lui-même. Le développeur examine chaque demande de fonction, vérifiant que le modèle est apte à satisfaire les utilisateurs. En général, les différentes fonctions basées sur le même modèle sont mutuellemnt indépendantes.

A l'issue de cette étape, il existe une spécification de chacune des fonctions à implémenter. Cette spécification est documentée de trois façons :
- une réorganisation de la demande informelle avec les termes du modèle initial;
- une élaboration du modèle initial SSD, montrant comment la fonction est construite soit dans un des processus du modèle, soit comment elle est connectée à un de ceux-ci;
- un texte en pseudo-code explicitant les spécifications détaillées de la fonction.

Le but de cette étape est de fournir les "output" demandés par l'utilisateur. De nouveaux processsus sont introduits, quand les processus du modèle ne sont pas suffisants pour satisfaire les requêtes des utilisateurs (function process).
Ils sont principalement de trois types :
- "embedded function" : celles-ci sont incluses directement dans un processus du modèle,
- "imposed function" : un processus est ajouté et il utilise un state-vector qui examine les variables d'un processus.
- "interactive function" : un nouveau processus est ajouté; celui-ci inspecte une state-vector d'un processus du modèle et réagit en produisant une suite de données vers le même processus qui l'utilise comme input. Il est préférable de créer un autre processus du modèle (LEVEL + 1) laissant le premier processus structurellement intact.

A noter que :
- Une fonction peut requérir plusieurs processus.
- Une fonction peut être connectée à plus d'un type de processus du modèle ou à un autre processus dédié à une fonction.
- Une fonction peut être interactive en regard d'une partie du modèle et "imposed" vis-à-vis d'une autre partie du modèle.

Pratiquement, il faut déterminer
1. quels processus sont nécessaires pour répondre aux demandes des "output" et comment elles sont connectées au SSD;
2. il est aussi nécessaire de spécifier le processus de chaque fonction et chaque modification des processus du modèle auxquels elle est connectée.

Règles importantes :
1. La durée de vie du processus (fonctionnel) doit être la plus large possible. Il ne peut jamais être spécifié en tant que procédure.
2. Il existe une limite à l'extension des processus du modèle aux fins de satisfaire les requêtes fonctionnelles. En effet, on ne peut modifier la structure du processus du modèle. Si une restructuration est requise, un nouveau processus du modèle doit être spécifié, connecté au processus originel par un "data-stream" (LEVEL-2 PROCESS). La seule chose que peuvent supporter les processus du modèle de niveau-1 (LEVEL-1) est l'adjonction de variables et d'opérations élémentaires sur celles-ci.
3. Il est déconseillé d'inspecter le pointeur de texte. Il faut le considérer comme une variable du système ; (un pointeur de texte est utilisé par entité pour situer l'état du processus par rapport aux actions qui vont se succéder et qui sont décrites dans le diagramme de l'entité).

Backtracking.
Principe connu : une décision a été prise alors que le résultat de l'hypothèse admise n'est pas encore disponible à ce moment.

Attribut et variable locale dans JSD.
La notion d'attribut cède le pas à celle de variable locale (le pointeur de texte n'est pas utilisé comme une variable du processus - il est préférable d'introduire une autre variable). En fait, il est possible et souvent habituel, d'anticiper les catégories générales des fonctions interrogatives et donc d'implanter bon nombre de variables locales d'usage général.
Il est aussi nécessaire d'introduire des variables qui deviennent partie intégrante du vecteur d'état mais qui n'ont pas de rapport direct avec l'exécution du processus et ne sont pas mis à jour par les actions des entités.
Ex : Nom, Prénom, Date de naissance, Adresse.....
De telles variables sont caractérisées par le fait que leur changements de valeur sont en dehors du modèle et ne sont pas coordonnés avec les actions des entités. Elles peuvent être représentées par un diagramme additionnel avec un nouveau type d'entité. Cependant le processus dégénère habituellement au stade de l'implémentation en une variable locale du type d'entité premier parce qu'il n'y a aucune fonction, dépendante de la valeur du pointeur de texte de ce processus additionnel et aucune connection à cette fonction si ce n'est pour l'inspection de ses seules variables locales.
Remarque :
Tout ce que l'on peut appeler un attribut d'une entité n'apparait pas

obligatoirement comme une variable de ce processus. Seule sont concernés dans JSD les variables locales de processus séquentiels qui modèlent les entités.

Résumé de l'origine des variables locales :
- le pointeur de texte est implicite dans le processus level-1,
- une forme (ou plusieurs) plus commode du pointeur de texte peut être ajoutée pour faciliter la liaison avec les fonctions,
- les attributs de l'occurence la plus récente d'un type d'action peuvent être retenus comme variable locale,
- la somme d'actions passées peut être calculée dans le but de satisfaire des fonctions associées.

5 SYSTEM TIMING STEP

Resume : il est nécessaire de prévoir les contraintes de temps de réponse et de délai pour les différentes parties du système. Celles-ci sont naturellemnt discutées avec les utilisateurs. Les décisions résultantes seront utilisés dans l'étape suivante où les choix d'implémentation devront satisfaire aux désidérata des utilisateurs.

Importante dans les systeme temps reel.
Introduction de processus de synchronisation . Ils synchronisent des processus en écrivant deux ou plusieurs TGM.

6 IMPLEMENTATION STEP

Résumé : a ce stade-ci, les spécifications sont :
- le SSD : il montre quels sont les processus et comment ils sont connectés l'un à l'autre et aux entrée-sorties du système (en fait les limites de l'application).
- Les spécification détaillées de chaque type de processus sont données sous la forme de texte structuré (+ diagramme).

Dans cette étape, il faut déterminer combien de processeurs réels ou virtuels seront utilisés par le système, l'allocation aux processeurs disponibles (quand ils sont moins nombreux que les processus), comment sont gérés les processus alloués un processeur. Par exemple, si le système doit être exécuté comme une simple tâche, il est nécessaire de le convertir à partir de cet ensemble de processus séquentiel eu un unique processus séquentiel. Les décisions concernant le "scheduling" sont prises en regard des informations provenant de l'étape précédente. Le schéma résultant est le System Implementation Diagram (SID). La subtilité de l'étape d'implémentation de JSD est que la

spécification du système, incorporé dans le SSD et les diagrammes des processus, est retenue dans l'implémentation. Le SID peut être vu comme une transformation du SSD. L'unique nouveau processus ajouté est celui du "scheduling" qui a été consciemment exclut des étapes de spécification.

JSD, dans ses spécifications, emploie beaucoup de processus. La tâche de l'implémentation sera de réduire le nombre de processeurs nécessaires.
Une manière est de simuler plusieurs processeurs nécessaires (multi-tasking...) en définissant des processeurs virtuels.
Une autre manière de réduire le nombre de processeurs est de transformer la spécification de telle manière qu'il y ait moins de processus séquentiels à exécuter. C'est une façon de faire purement liée à cette étape. Cela ne change en rien le nombre de processus spécifié antérieurement. On peut faire en sorte que pour la machine, il apparaisse un seul processus, alors qu'en fait c'est un ensemble composite de plusieurs processus.
1. Transformation avec scheduler.
Technique de l'inversion d'un processus par rapport à un input (data-stream) :

passage d'un processus séquentiel à une subroutine qui manipule un record à chaque appel. Le mécanisme de la subroutine est utilisé pour suspendre (CALL) et reprendre (RETURN) l'exécution d'un processus. Souvent l'implémentation avec LABEL et GOTO est considérée comme mauvaise mais il faut concevoir les processus inversés sous l'angle de code-objet et non de code-source.

2. Transformation sans scheduler.
Un des processus joue le rôle de scheduler implicitement.

A) Conditions pour utiliser la technique de combinaison de processus par inversion sans ajouter un processus scheduler :
- seul des data-stream sont utilisés,
- pas de rough merge dans la combinaison,
- chaque couple de processus du système est connecté uniquement par un seul chemin.
NB : n'importe quel processus peut être choisi comme le schéduler effectif en sachant l'influence que cela aura.

Remarques importantes.
- chaque processus assigne des valeurs définies à chaque partie de son vecteur d'état au commencement de son texte, avant l'exécution de chaque opération de communication avec chaque autre processus,
- la création d'un processus doit être regardée comme l'exécution du processus au moins aussi loin que la première opération de communication avec d'autres processus,
- le vecteur d'état d'un processus ne peut pas être inspecté tant que le processus de création (comme défini plus haut)n'est pas complet,
- l'exécution d'un système est précédée par la création de tous les

processus qui doivent être présents initialement (ils ne sont pas donc créés par aucun autre processus).

B) Extension de la combinaison de processus par inversion au "rough merge" :
On peut adjoindre un processus avec "rough merge" dans un ensemble de plusieurs processus inversés à condition qu'il accepte l'ordre des "record" produit par les processus inversés comme le résultat du "rough merge" dans ce même processus quand il n'est pas encore intégré. Evidemment l'inversion en fait une subroutine des processus combinés antérieurement.

C) Channel
Un channel est une connection entre un processus scheduler, un processus X par exemple, par rapport auquel sont inversés deux ou plusieurs "data-stream" (qui proviennent de deux ou plusieurs processus connectés au scheduler) Le scheduler doit être capable. de savoir à quel endroit le processus X est suspendu pour pouvoir appeler un autre processus inversé. Une opération query permet de le déterminer. L'inversion nécessaire pour donner une connection de type channel est un peu plus complexe que l'inversion avec respect à un simple data-stream.
Remarques importantes.
1. Pour produire les informations nécessaires requises pour chaque opération d'interrogation (query), le processus inversé doit exécuter chaque segment en amont du processus appelant. A ce moment seulement le premier query peut être effectif.
2. L'imbrication du processus appelant et appelé dans une inversion simple dépend du data-stream mis en cause par l'inversion (input ou output). Un conflit peut surgir quand il y a inversion par rapport à l'input et à l'output à la fois. Il peut être résolu quand on introduit un élément de stockage (buffer) sous la forme d'un record additionnel pour l'un des data-stream et en y ajustant l'imbrication comme requise.
3. Si un simple espace dédié à un record est utilisé par un data-stream et est passé comme paramêtre par le processus appelant, alors la zone doit être gardée intact aussi longtemps que nécessaire par le processus appelé.
4. Possibilité d'inverser un processus par rapport à un autre et d'inverser la combinaison par rapport à deux ou plusieurs data-stream. Un certain soin est nécessaire pour implementer ces opérations (un pointeur de texte pour le processus combiné et un dans chaque processus pour la combinaison).
5. L'implémentation des channels est grandement simplifié dans certains cas. Le plus important est celui où le séquencement des opérations est parfaitement prédictible (l'opération query n'est plus nécessaire).

Scheduling avec buffer.
Les buffers sont placés sous le contrôle du scheduler et accessible seulement à celui-ci (principalement quand il y a data merge).

Démembrement d'un processus (dismembering).
Quand un processus est inversé par rapport à deux processus distints, on peut alors inversé une version simplifié du processus à chacun des deux processus. Chaque version ne gardera les actions qui concerne le processus par rapport auquel elle est inversée.

Data-base design et implémentation.
Une base de donnée dans JSD est vue comme une collection de state-vector des processus du système (processus du modèle, des fonction et quelques fois du scheduler). Le contenu de chaque vecteur est déterminé par les besoins du processus et quand il y a connection par state-vector par les informations nécessaires aux autres processus qui inspectent le vecteur. A noter que les vecteurs peuvent être partagés comme les processus (dismembering).
Le pointeur de texte est explicite dans chaque vecteur d'un processus. Le rôle joué par le pointeur n'est pas retrouvé dans le shéma typique d'une base de donnée. En fait, ce rôle est assigné à la combinaison d'autres variables qui jouent ce rôle en plus de leur propre rôle.

Rétroparcours.

Les six étapes doivent être parcourues dans l'ordre donnée. Les décisions prises dans chaque étape définit le contexte des décisions pour les étapes suivantes. En pratique, quelques itérations sont parfois nécessaires : oubli d'une entité, oubli d'une action, impossibilité de spécifier une fonction en terme des entités et actions modelisées.

Conclusion

M.Jackson se justifie clairement. "Une manière de caractériser une méthode est de spécifier de quelles façons elle va peser sur les décisions lors du developpement (en précisant que décision est pris au sens large du terme). Chaque type de décision a ses propriétés dépendantes de leur nature intrinsèque et leur place dans l'organisation des décisions de la méthode.
Principes méthodologiques auxquels satisfait JSD :

Les décisions les plus faciles doivent être prises avant les plus difficiles.
Les décisions les plus sujettes à erreur doivent être reportées le plus longtemps possible.
Les décisions implicites sont à éviter.
Si une décision est susceptible d'entrainer une erreur, il doit être possible de confirmer ou de la refuter le plus vite possible.
Autant que possible, les décisions seront indépendantes l'une de l'autre".

L'auteur de cette méthode se défend bien d'assimiler JSD à une méthode top-down. L'idée fondamentale du top-down est la hierarchie. Celle-ci peut être une hiérarchie de subroutine, de diagramme ou de flux de données. C'est une méthode qui correspond à une façon raisonnable de décrire les choses qui sont completemment comprises. Elles n'est pas une méthode rationnelle pour développer, structurer ou découvrir quelque chose. Il ne faut pas confondre méthode de description et méthode de développement. La méthode top-down est surtout valable quand le problème est résolu. Une méthode de développement doit pouvoir résoudre des problèmes dont il(le développeur) ne connait pas la solution, de développer un système qu'il ne peut pas se représenter d'un seul coup d'oeil. A ce point de vue, top-down est la plus mauvaise méthode car elle contraint le développeur à prendre au début du dévelopement les décisions les plus larges et qui portent le plus de conséquences.
Elle semble moins fastidieuse à utiliser que la méthode top-down. Elle ne définit pas les fonctions de façon détaillé (formattage des report, saisie des données,...). A ce point-là, quand le développement est fini suivant cette méthode, la méthode de décomposition fonctionnelle peut très bien prendre le relais pour ce coté plus lié à la programmation stricte (définir des primitives d'affichage,..).

Remarquons que :
L'indépendance des fonctions est importante à la fois dans le développement originel du système où il permet à plusieurs personnes de travailler simultanement sur l'étape "function step" et dans sa maintenance durant l'utilisation productive, où il permet à de nouvelles fonctions d'être ajoutées sans interférer avec les précédentes.
Un prototype d'un système peut être construit facilement.

Le développement d'un système de manière progressive est grandement facilté par cette méthode.

APPENDIX B

INFORMATISATION DES DONNEES MEDICALES.

B.1 INTRODUCTION

La présente annexe propose une analyse possible en vue d'une informatisation des données médicales associé aux dossiers médicaux.

B.2 LE PROTOCOLE DE TYPE TEXTE

Les diagnostics précis quant à la maladie et à sa localisation que les médecins posent s'établissent largement grâce aux examens techniques. Le rôle que ceux-ci jouent est primordial. La multitude des examens et le caractère régulier de certains a augmenté ces dernières dans des proportions considérables. Le but de ce travail n'est pas de mettre en cause l'usage excessif ou non des investigations techniques mais de se préoccuper de la gestion que la diversité et la multiplicité entraînent. Aussi ce paragraphe met l'accent sur la production des données médicales, issues des services d'examen technique. Le protocolage devient le point de départ d'information structurée et utilisable.

Les protocoles de type texte, complet c-à-d où pas (peu) de valeur quantifié par l'appareil lui-même interviennent, présentent plusieurs problèmes à la gestion des données médicales qu'ils contiennent.Les examens radiographiques et endoscopiques sont complètement liés à celui qui les protocole.
D'autres examens s'expriment sous une forme facilement manipulable mathématiquement mais leur interprétation doit prendre en considération des éléments extérieurs à ceux-ci. Le protocolage d'un électrocardiogramme est un exemple de ce type d'examen.
De nouveaux critères de structuration doivent permettre de les traiter dans un système d'information.

B.2.1 Utilisation des examens techniques

L'utilisation des examens techniques est double.

La première fonction, immédiate, est le diagnostic et le contrôle

Le médecin estime que la confirmation d'une hypothèse nécessite un ou plusieurs examens techniques. Dans le cas d'un patient ulcéreux, un examen radiographique objective la présence d'un ulcère. Une prise de sang confirme de la même manière un diabète, une hépatite. Beaucoup de pathologie mettent un temps plus ou moins long à guérir ou bien sont chroniques. Le diabète, maladie endocrinienne incurable, demande de nombreux contrôles afin d'éviter les complications et les risques supplémentaires qu'apporte le déficit d'insuline. L'exemple du pneumothorax dont il s'agit de suivre l'évolution jusqu'à résorbption complète (une radio tous les deux jours) montre bien la fonction de contrôle qui est complémentaire de la fonction diagnostic. Un autre exemple courant : les examens préopératoires. Ils regroupent quelques examens systématiquement fait à titre de prévention avant toute anesthésie. Ils éliminent un certain nombre de risques dû à ce manque d'informations. Pour mémoire, ces examens sont classiquement une prise de sang, un électrocardiogramme, un avis cardiologique, une radio du thorax et les épreuves respiratoires. Cinq protocoles qui la plupart du temps sont normaux ou du moins révèlent rarement une anomalie méconnue. Ces protocoles rejoignent le dossier médical où ils sont insérés comme expliqué dans le premier chapitre.

La seconde fonction, plus lointaine, est la recherche médicale

Actuellement, ce volet fait partie intégrante de la médecine et tout problème médical quelque soit son évolution et son issue, reste une source d'enseignement. Les protocoles, faisant partie intégrante des dossiers médicaux et se basant sur des faits objectifs, de plus en plus normalisés sont une mine d'or pour les générations futures; la recherche dans ce domaine est faite et par les médecins d'un service pour chercher par exemple tous les protocoles de malades ayant une affection (pas nécessairement en relation directe avec le type d'examen) et par les médecins protocoleur qui veulent mettre en évidence certains points sur un même type d'examen. Ces aspects, spécialement dans les grands hôpitaux toujours universitaire ou para- universitaire chez nous représente une part qui grandira dans l'avenir.

B.2.2 Analyse des protocoles de type texte

Ce sont particulièrement ceux-ci qui méritent l'attention de l'informaticien. Une analyse plus minutieuse du protocole (de type texte) à travers les impératifs auquels il doit répondre, dégage une structure plus aisément utilisable par un système d'information.
Le protocolage endoscopique offre la complexité la plus forte.

Structure résumée du PROTOCOLE (type texte).

1. DONNEES SIGNALETIQUES

2a. TYPE D'EXAMEN
2b. APPAREIL UTILISE
3. SYMPTOMES PRINCIPAUX
4. PROTOCOLE PROPREMENT DIT
   A. CORPS
   B. CONCLUSION
5. EXAMEN(S) COMPLEMENTAIRE(S)
6. TRAITEMENT PROPOSE
7. COMMENTAIRE

Les investigations de radiographies et autres ne possède pas toute les rubriques définies ci-après.

Les rubriques 1, 2a et 2b n'offrent pas beaucoup de difficultés. Les données signalétiques doivent être introduites dans la machine. Le type d'examen, l'appareil utilisé et les examens complémentaires font appel à un choix très limité où un ou deux menus aident à l'introduction de ces informations.
La rubrique (7) COMMENTAIRE, permet au médecin d'insérer quelques lignes de texte libre. Comme elles sont individuelles, leur encodage doit se faire par le médecin lui-même ou la secrétaire. Elles ne peuvent pas utiliser la méthode de choix successifs qui est développé pour les autres rubriques.
Les rubriques restantes
   3. Symptômes principaux
   4. Protocole - A corps
              - B conclusion
   6. Traitement
contiennent les données médicales les plus difficiles à normaliser (cfr aussi annexe D).
   Remarque : la composition et l'ordre dans lequel les éléments sont cités dépendent du médecin-protocoleur. Il est son propre guide. Chaque médecin, maintenant, enregistre l'interprétation de l'examen, suivant son humeur, sa fantaisie, le service où il a été formé... Ce qui veut dire que deux médecins qui protocolent un même examen avec le même diagnostic sur un même patient ne le rédigent pas de la même façon. Cette manière de faire empêche tout traitement de ces protocoles pour quelques decennies encore. Pour pallier à ce problème, il semble logique de proposer une succession de menus. De cette façon, le médecin protocoleur se laisse diriger dans un arbre reprenant de façon systématique toutes les possibilités existantes ou connues. Ce procédé laisse la liberté de choix au médecin et contourne le principal écueil que la manière actuelle de protocoler ne sait éviter. Aussi réaliser des arborescences complètes est difficile mais garantit une systématisation rigoureuse. Quand une séquence de mot-clef est sélectionnée, par menus successifs, elle définit soit un descriptif des plaintes ou de l'organe examiné, soit la conclusion ou le traitement. Le rôle du médecin est de déterminer dans tous les menus qui se succède à l'écran, les mots ou expression qui correspondent à ce qu'il a vu de façon univoque. L'élaboration de chaque noeud de l'arbre demande bien du travail et de l'affinement mais il est réalisable après quelques efforts. Précisons qu'il faut nettement définir les deux parties du protocole proprement dit. Le corps doit rester descriptif et ne comporter aucun

élément diagnostic. Celui-ci ne sera abordé que dans la conclusion. Actuellement, les protocoles ne font pas nécessairement la destruction entre partie descriptive et partie conclusive. D'autre part, si le traitement ultérieur de ces informations devient possible grâce à une structure bien établie (en l'occurrence, l'arbre), les médecins devront faire l'effort d'envisager le protocolage comme un parcours logique et rationnel qui ne tolère pas de fantaisie.
On peut opposer que la séquence de mot-clef ou expression n'est pas un texte français comme le protocole l'était avant. Cet handicap n'est qu'apparent et un système de composition de lettre peut à partir de cette succession de mot-clef retraduire ceux-ci en un texte normal. La conclusion du protocole pourrait être déduit de la description et laisser le choix à quelques éventualités surtout quant au facteur de certitude (affection vraissemblable, certaine,...).

B.2.3 Utilisation du protocole dans le dossier médical

La taille d'un protocole de type texte varie suivant le type d'examen et aussi l'affection découverte par l'examen. Un protocole de type chiffre a de toute façon une longueur identique à la demande d'examen. Tout paramètre est déterminé. Par contre, un poumon radiographié, s'il est normal n'éveille aucun commentaire particulier. Une lésion découverte, au cours de ce même examen, sera décrite dans le détail et allongera le protocole. Que retient le médecin dans son dossier médical de ce protocole d'examen qu'il a requis ?

En règle générale, il recopie deux ou trois lignes qui contiennent ce qui est intéressant dans le contexte du patient. Le protocole complet est gardé dans le dossier, pour consultation ultérieures plus détaillées si nécessaire. On comprend que pour le médecin recopier tout le protocole sur les feuilles de travail ne sert à rien puisque celui-ci se trouve en fin de dossier.

B.3 LE DOSSIER MEDICAL INFORMATISE.

B.3.1 Introduction.

Des bases de données médicales existent déjà. Une des plus importantes sont les registres médicaux. Ils sont constitués à partir d'un masque d'entrée. La présence ou non de tel examen, de tel symptôme, de tel traitement valide l'indicateur correspondant. Si dans le cadre d'une maladie particulière, il peut se révéler très utile, pour un système d'information qui se veut le plus général possible, il faut adopter une base de données plus souple. Une analyse des éléments impliqués dans les feuilles de travail s'impose. Les lignes suivantes se proposent de schématiser les rubriques et par la même de se plier à toute évolution médicale. Cette approche (qui n'est certes pas la seule) tente de rendre réalisable une gestion complète des données médicales et essaye de mettre en évidence les points délicats.

B.3.2 Analyse des rubriques

B.3.2.1 Antécédent

SCHEMA PROPOSE

ANTECEDENT
familiaux
    PARENTE
    AGE
    MALADIE
    DECES
personnels
    PROBLEME DE SANTE
    AGE(DATE)
    TRAITEMENT(S)

Mémoire des principales étapes du vécu médical du malade jusqu'à la création du dossier ou entre deux hospitalisations (ou consultations) successives, un autre schéma habituel comme ci-dessous ne prend pas bien en compte les concepts d'antécédent et ne traduit pas correctement l'historique médical.

antécédent
- familiaux
  ....
- personnels
  . médical
  . chirurgical
  . psychique

Le point principal est la pathologie et son retentissement sur la personne atteinte. Si une personne, par exemple, a quelques épisodes d'ulcère gastrique et guérit plusieurs fois avec un traitement médical, il arrive souvent qu'une sanction chirurgicale soit prise pour la même pathologie.

Exemple

```
      antécédents
          - familiaux
              ...
          - personnels
              médical
            ULCERE GASTRIQUE  MAI  79   TAGAMET
            ULCERE GASTRIQUE  OCT  80   DE NOL
              chirurgical
            APPENDICECTOMIE   JUIN 6
            ULCERE GASTRIQUE  AVR  82   VAGOTOMIE
                                          SUPRA-SELECTIVE
              psychique
            DEPRESSION        NOV  81   LERIVON
```

La chronologie n'est plus vérifiée et le problème se retrouve dans deux, peut-être dans les trois subdivisions (la dépression est vraisemblablement une conséquence du stress intense et ,en même temps, à l'origine de l'aggravation de l'ulcère vers la perforation). Mr WEED suggère l'utilisation de "record médical" en quelque sorte et dans ce cas respecte beaucoup plus le concept de maladie en tant qu'unité.
L'exemple se modifierait comme suit :

```
        antécédent
            familiaux
               ...
            personnels
               APPENDICECTOMIE   JUIN 67
               ULCERE GASTRIQUE  MAI  79  TAGAME
                                 OCT  80  DE NOL
                                 AVR  82  VAGOTOMIE
                                            SUPRA-SELECT.
               DEPRESSION        NOV  81  LERIVON
```

La chronologie dans la maladie est respectée mais il me semble préférable de faire prévaloir (avis personnel) la chronologie des problèmes sur la l'historique médical dans une pathologie déterminée (l'ulcère ici).
L'exemple devient alors :

```
      antécédent
          familiaux
             ...
          personnels
            APPENDICECTOMIE   JUIN 67
            ULCERE GASTRIQUE  MAI  79  TAGAMET
                              OCT  80  DE NOL
            DEPRESSION        NOV  81  LERIVON
            ULCERE GASTRIQUE  AVR  82  VAGOTOMIE
                                         SUPRA-SELECTIVE
```

Bien qu'émaillée par d'autres problèmes de santé, la dynamique et la continuité de la vie médicale du patient ressort nettement. Il faut préciser que l'appendicectomie est un traitement qui en lui-même précise la pathologie. Certains termes médicaux principalement en rapport avec une

opération ont une sémantique plus large. Le terme de splénectomie (enlèvement de la rate) par contre devra être accompagné d'un mot justifiant cette intervention, les causes étant variées.

B.3.2.2 Données subjectives ou anamnèse

ANAMNESE	SCHEMA PROPOSE ANAMNESE
SYMPTOME	(1)
TYPE	(2)
CARACTERE	(3)
LOCALISATION	(4)
PERIODICITE	(5)
DEBUT	(6)
DUREE	(7)
MODALITES	(8)

Un symptôme est un phénomène qui révèle un trouble fonctionnel ou une lésion. Il peut être objectif ou subjectif. Dans ce cas-ci, le terme symptôme est pris dans le sens subjectif, ressenti par le patient. La liste des symptômes n'est pas limitative. Vouloir les déterminer tous est un travail gigantesque sans oublier que les termes sans oublier que les termes sous des noms similaires (franglais ?!) n'ont pas toujours la même signification pour des écoles différentes. Chaque spécialité a son vocabulaire propre. La psychiâtrie est un exemple de la complexité des symptômes à établir et à normaliser.
Les attributs mentionnés ne sont pas toujours tous nécessaire mais ils suffisent dans la grande majorité pour caractériser toute anomalie ressentie par le patient.
Exemple :

- Fatigue(1) 2mois(6)

- Douleurs(1) crampoïdes(2) violente(3) épigastre(4)
  1mois(6) 1/2h(7) 1h postprand.(8)

- Céphalée(1) frontale(4) 3-4x/j(5) 15min(7) 5sem(6)
  vomissements(8)
  vertiges(8)

L'attribut modalité reste le plus vaste et le plus discutable. Une analyse supplémentaire pourrait le détailler plus. On peut se poser la question : est-ce que ce sont les vomissements qui sont accompagnés de céphalées ou bien, comme supposé dans l'exemple, la céphalée responsable des vomissements et vertiges ? Quand la réaction de cause à effet se détermine facilement et que de plus la symptomatologie des vomissements et vertiges se superpose facilement, il n'y a pas de problème. Mettre les trois symptômes dans le premier attribut traduit bien cette appréhension du réel. Dans le cas où le doute est permis, la solution consiste à séparer les deux phénomènes en créant une nouvelle occurrence de symptôme avec des nouvelles valeurs attribuées aux autres items.

Exemple :

- Céphalée(1) frontale(4) 3-4x/j(5) 15min(7) 5sem(6)
  vomissement
  vertiges

B.3.2.3 Données objectives ou examen clinique.

SCHEMA PROPOSE

EXAMEN CLINIQUE
ANOMALIE ou DENOMINATION DE L'EXAMEN
VALEUR (CHIFFREE)
LOCALISATION
TYPE
EVALUATION (SYMBOLIQUE)

Par son examen, et suivant sa spécialité, le praticien met en évidence plusieurs irrégularités chez une personne. La diversité des symptômes objectifs équivaut bien à celle des données subjectives. Le jargon médical regorge de tests et de signes particuliers. Les spécialités comme la neurologie occupe une place de choix quant à leur nombre et à leur nom (qui est bien souvent celui de leur auteur). Il est important de signaler une différence par rapport aux données recueillies lors de l'anamnèse. En supposant que la collection des plaintes (anamnèse) soit totale et correcte dans un cas donné (ce n'est pas le plus évident), le médecin est assuré de connaître tout ce qui l'intéresse. Au niveau de l'examen clinique, le patient n'est pas toujours conscient d'une anomalie qu'il présente. Le médecin recherche les données objectives et fait une démarche intellectuelle qui le guide vers certains tests ou examens. Une donnée de l'examen clinique "normale" doit être pris en compte car la connaissance de celle-ci apporte beaucoup plus de renseignements que le fait subjectif, considérée comme normal et donc, non retenu. Le malade est très rarement conscient, p.ex. que d'un côté son réflexe rotulien est normal alors que de l'autre, il apparaît augmenté. Dans l'évolution et le suivi d'une pathologie, la référence à un examen clinique antérieure complet s'avère primordial dans bien des cas. La manière de consigner ces résultats dans le dossier varie. A côté de valeur chiffrée, il existe une classsification de type symbolique ou de type tout ou rien.

Exemples :

TA	15/9	BASE DROITE	RALES HUMIDES	++
	2/6	FOYER MITRAL	SOUFFLE SYST.	
	1/8		EXTRASYSTOLE	

La sémantique médicale ici s'affirme diffuse. Le terme "base droite" localise sans aucun doute le poumon comme organe examiné. De même que "foyer mitral", "souffle systolique" suggèrent sans ambiguité une auscultation cardiaque.
D'autres exemple :

OEDEME	MEMB. INFERIEURS	RETENTION	++

TUMEUR SOUS-CUTANE LIPOME +

Le médecin doit s'efforcer de rester descriptif et ne pas inclure de notion diagnostic, si possible. Bien souvent, un diagnostic est implicitement évoqué mais il ne peut être relaté dans cette rubrique.
L'exemple suivant d'une fracture :

CONTUSION 1/3 SUP TIBIA G. +++
DOULEUR

Dans un cas aussi évident, ou du moins aussi suspect, le diagnostic sera posé sur radiographie et ne nécessite aucune description au point de vue clinique. Préciser la douleur que l'on peut deviner comme étant exquise, présente peu d'intérêt. La traumatologie se prête difficilement à une systématisation compatible avec les autres pathologies.

B.3.2.4 Demande d'avis

SCHEMA PROPOSE

DEMANDE D'AVIS
EXAMEN (TYPE)
MOTIF

La demande est très simple. Le seul problème est que la liste des motifs est illimité et que parfois un texte est nécessaire. Dans le dossier médical, le motif de la demande se retrouve dans les symptômes objectifs ou/et subjectifs; il n'y a pas nécessité de le marquer en plus sur les feuilles de travail. Le médecin par contre a besoin de ces renseignements bien souvent.
Exemple

ORL DL OREILLE DROITE

B.3.2.5 Demande d'examen.

SCHEMA PROPOSE

DEMANDE D'EXAMEN
EXAMEN (TYPE)
MOTIF

même remarque que ci-dessus.
Exemples :

RX THORAX DOUL.BASE DR.
RALES HUMIDES

B.3.2.6 Réponse d'avis

SCHEMA PROPOSE

REPONSE D'AVIS
- DIAGNOSTIC
- TRAITEMENT
- EX COMP.

Il contient quelques mots sur le diagnostic et parfois une suggestion de traitement ou d'examen complémentaire.
Exemple :

PSORIASIS LOCACORTENE BIOPSIE

B.3.2.6.1 Protocole

SCHEMA PROPOSE

PROTOCOLE
- EXAMEN
- MOTS-DIAGNOSTIC
- TRAITEMENT
- EX. COMPLEMENTAIRE

Ils reprennent une partie des éléments contenus dans le protocole vu antérieurement. Un radiologue ne propose pas de traitement mais peut-être bien un examen complémentaire. Un support image s'associe souvent à ce protocole.
Exemple :

ECG INFARCTUS POSTERIEUR SINTRON
XYLOCAINE

B.3.2.7 Diagnostic/traitement

SCHEMA PROPOSE

DIAGNOSTIC/TRAITEMENT
- PROBLEME DE SANTE
- TRAITEMENT
- POSOLOGIE

Pour pouvoir répondre à la démarche médicale, cette rubrique doit comporter et le diagnostic et le traitement. La proposition de "record" de WEED s'adapte parfaitement à ce problème. Une suite de données subjectives ou objectives du patient, associée à des résultats d'avis et d'examen conduisent le practicien à un ou plusieurs diagnostics. De chacun de ceux-ci, s'il y en a plusieurs, découlera une prescription de médicaments ou un traitement chirurgical. Une série de perturbations décelées ou confessées par le patient lui-même peut ne pas avoir trouvé de diagnostic dans un premier temps et néanmoins répondre à un thérapeutique.
Exemple :

PNEUMONIE VIBRAMYCINE 3X/J

B.3.2.8 Lettre au médecin-traitant

SCHEMA PROPOSE

LETTRE TRAITANT
TEXTE

Ce document est capital pour deux raisons. Il résume toute l'hospitalisation ou le(s) consultation(s) et explique la démarche propre du médecin, le fil conducteur qu'il a suivi. Dans le cadre de données informatisées, elle prend une valeur particulière parce qu'elle est le seul élément qui reste en texte libre, dicté comme il était auparavant. Au côté aride, structuré des dossiers médicaux informatisés s'oppose ce morceau souple de texte personnel. Tout examen cité, toute opération dont on parle pourront accompagner cette lettre, si le rédacteur l'estime nécessaire, en joignant un exemplaire des protocoles en annexe. La lettre au médecin-traitant n'apporte rien de neuf en ce qui concerne les données médicales et doit de ce fait être un intermédiaire entre deux confrères où se mêle aussi un peu de sensibilité humaine. De plus, elle permet de glisser toute une partie non automatisable pour le moment (la stratégie employée, quelques problèmes techniques, choix d'un médicament par rapport à un autre). Elle devient un aide mémoire pour celui qui l'a composée. L'impression des informations à l'état brut laisse la place à une ou deux pages de prose. L'automatisation à l'extrême, n'a aucune justification pour elle-même.

B.3.2.9 Problème non résolu

SCHEMA PROPOSE

PROBLEME NON RESOLU
identique à
DONNEES OBJECTIVES
et
DONNEES SUBJECTIVES

Si en général, la plus grande partie des plaintes et des désordres tant objectif que subjectif sont résolues, à la grande satisfaction du patient, comme on peut l'espérer, quelques troubles que le médecin n'arrivent pas à éliminer, subsistent. Cette rubrique calquée sur les deux précédentes (données objectives et données subjectives) isole ces points d'interrogation.
Exemple :

DOULEUR FID LEGERE CONSTANTE 2MOIS AUGMENTE A L'EFFORT

B.3.2.10 Cause(s) principale(s)

SCHEMA PROPOSE

CAUSE PRINCIPALE
identique à
DONNEES OBJECTIVES

et
DONNEES SUBJECTIVES
MOTIF(non médical : accident, tentative de suicide,..)

Si la rubrique diagnostic/traitement se compare à la conclusion, on ne peut oublier de spécifier cette autre donnée utile à la recherche médicale : cause(s) principale(s) de la (les) consultation(s) ou hospitalisation. Cette rubrique reprend les mêmes attributs que pour l'anamnèse et/ou l'examen clinique.
Exemple :

VOMISSEMENT SANGLANT
ACCIDENT

B.3.2.11 Protocole opératoire.

Dans un premier temps de type TEXTE, il pourrait bien progressivement s'automatiser comme le protocole d'examen technique de type-texte. Cependant les difficultés à établir les noeuds de l'arbre ne sont pas négligeable parce que bon nombre d'imprévus doivent être prévus (?!). Il me semble utile de le résumer en quelques mots sur les feuilles de travail et de le rappeler si nécessaire à la mémoire en le lisant.

SCHEMA PROPOSE
PROTOCOLE OPERATOIRE
OPERATION (TYPE)
NOM
MOT-CLEF

B.3.2.12 Complication

SCHEMA PROPOSE
COMPLICATION COMPLICATION
CAUSE
TYPE

Toute pathologie provoque des complications dans un petit(!) pourcentage de cas. Un traitement, qu'il soit médical ou chirurgical, peut aussi amener des complications. Un examen technique sanglant (p.ex un artériographie, une coronarographie...) cause un certain nombre de problèmes secondaires à cet examen. Aussi, il est difficile de ne pas prévoir une rubrique qui de temps en temps sera rempli dans le dossier et permettra de faire, plus tard dans la recherche, de fructueuses constatations.
Exemple :

APPENDICECTOMIE PERITONITE
CHLORAMPHENICOL APLASIE MEDULLAIRE

B.3.2.13 Fiche médicale

SCHEMA PROPOSE

Fiche-résumé
DATE DE MISE-A-JOUR
Historique
  PROBLEME MEDICAL
  DATE
  DATE(fin d'hospitalisation)
  TRAITEMENT
Affection actuelle
- PROBLEME MEDICAL
  TRAITEMENT
  POSOLOGIE
- ANOMALIE ou DENOMINATION DE L'EXAMEN
  VALEUR (chiffrée)
  LOCALISATION
  TYPE
  EVALUATION (symbolique)
- SYMPTOME SUBJECTIF
  TYPE
  CARACTERE
  LOCALISATION
  PERIODICITE
  DEBUT
  DUREE
  MODALITE

Exemple :

Affection actuelle
  ALLERGIE PENICILLINE
  ANCOR CEDOCARD 3x 5mg/j
  DOULEUR ANGLE SPLENIQUE ?

B.4 ASPECTS ORGANISATIONNELS DE L'ACCÈS AUX DONNÉES MÉDICALES

B.4.1 Introduction

Un patient est une personne physique qui reçoit des soins médicaux (à titre diagnostic ou thérapeutique) d'un médecin ou de son équipe. L'équipe médicale comprend le plus souvent, un médecin-chef, un médecin-chef adjoint, un ou plusieurs assistants affectés à cet équipe(junior ou senior), un ou plusieurs stagiaire designé pour ce service. Parfois, il y aussi quelques résidents en plus. Le staff infirmier et du secrétariat fait partie de l'équipe médicale. Cependant, comme il n'est question d'aborder cet aspect que dans ses grandes lignes (le sujet mériterai certainement un mémoire à lui tout seul), seul le staff médical proprement dit sera pris en compte.

B.4.2 Situation actuelle

Tout médecin peut avoir accès aux dossiers médicaux de tous les patients. En règle générale, le secrétariat qui s'occupe de la réservation des lits du service et des rendez-vous, si le dossier existe déja, demande aux archives de le faire parvenir la veille ou le matin de l'entrée du patient ou de son rendez-vous. Inversement, tout dossier dont la lettre au médecin-traitant a été vérifié par le medecin retourne aux archives. Le secrétariat occupe une place privilégiée dans les mouvements du dossier. Quand une recherche est effectuée, les références du dossier sont sélectionnées à partir d'un index (qui est souvent déja informatisé) et le secrétariat gère leur acheminement pour les membres de l'équipe qui procède à cette recherche. Tous les abus restent possible mais la déontologie médicale est assez stricte.

B.4.3 Situation nés de l'informatisation des données

Si l'on suppose les trois étapes précedemment décrites réalisées, le dossier médical complet et la fiche-résumé de tous les patients se trouvent introduits dans le système d'information de l'hôpital. La situation offre quelques différences.
Le dossier-papier est unique. Pour prendre connaissance des informations médicales qu'il contient, sa disponibilité physique est nécessaire. Le dossier informatisé peut être disponible en même temps à plusieurs endroits.
Le dossier-papier prend un certain temps, pour transiter dans l'hôpital. La distribution par le canal informatique est rapide, si pas immédiate et la demande se fait à un terminal (donc ne requiert plus obligatoirement de passer par le secrétariat.

B.4.4 Le dossier médical complet

Des restrictions afin d'éviter des abus sont nécessaires. Elles concernent au premier chef, le dossier médical qui, dans le modèle exposé plus haut, contient beaucoup de renseignements à caractère privé (mode de vie...).

Proposition :

1. Un dossier informatisé n'est accessible qu'à des points d'accès définis pour celui-ci (le service concerné plus le bureau du chef de service, par ex.).

2. L'accès aux données médicales du patient produites pendant une hospitalisation est sous la rsponsabilité du chef de service.

3. L'accès aux données médicales du patient produite antérieurement à son hospitalisation reste à disposition du médecin (ou son équipe) qui les a enrégistrées, mais avec conservation de la trace de la demande. Les données du patient, provenant d'un autre service (exception faite d'un changement de service au cours d'une hospitalisation) ne sont accesible interactivement que après en avoir fait la requête au service d'archivage. Les données ne sont imprimables que par le secrétariat du service d'archive.

4. Toutes les données collectées sur des questions d'ordre privé (notamment le questionnaire d'ouverture et ses mises-à-jour) ne sont accessible qu'anonymement et dans le cadre d'une recherche scientifique controlé par le responsable. Il serait utile de définir un statut de la recherche scientifique en matière d'accès aux données.

5. Les données des patients d'un autre service ne sont pas accessible dans les circonstances normales.

B.4.5 La fiche-résumé

Son utilisation est double : interne et externe.

Proposition.

1. Dans l'hôpital, tout médecin a accès à la fiche des patients auquels il a prodigué des soins.

2. Le médecin-chef de service des urgences a accès à la fiche-résumé de tous les patients de l'hôpital

3. A l'extérieur de l'hôpital, la divulgation de la fiche-résumé d'un patient ou sa transmission par un réseau vers un autre hôpital est sous la responsabilité du chef de service des urgences de

l'hôpital qui possède ces données.

B.4.6 Rôle du service d'archive

Le service d'archive dans l'exemple d'une organisation de dossiers informatisés, joue un rôle central dans tout l'hôpital. Tout accès à des données médicales sera contrôlé par ce service qui possède finalement la gestion de la plus grande quantité de celle-ci. Le programme de recherche sera effectué par ses soins à la demande des chefs de cliniques. Si jamais des abus se produisaient, une consignation des accès permettrait de déterminer les contrevenants. Les services des examens techniques auront toujours à diposition leur données.

B.4.7 Rôle du service central

Comme tout patient qui rentre dans un hôpital, passe obligatoirement par les entrées, il semble utile que ce soit ce service qui définissent les points d'accès du dossier médical en envoyant cette information au service d'archive.

B.4.8 Conclusion

Ceci demande encore bien des études tout comme l'informatisation elle-même mais il serait hasardeux de ne pas prévoir les abus possible.

Facultés Universitaires Notre-Dame de la Paix

ANALYSE EN VUE D'UNE INFORMATISATION DES DONNEES AU NIVEAU HOSPITALIER

(ANNEXE GRAPHIQUE)

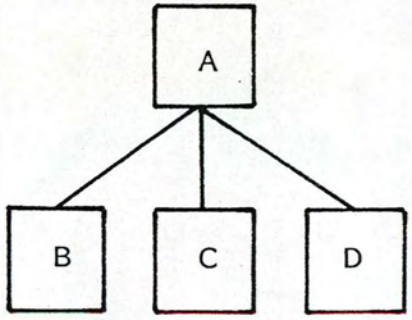
Promoteur : Jean FICHEFET

Mémoire présenté en vue de l'obtention du grade de maître en informatique

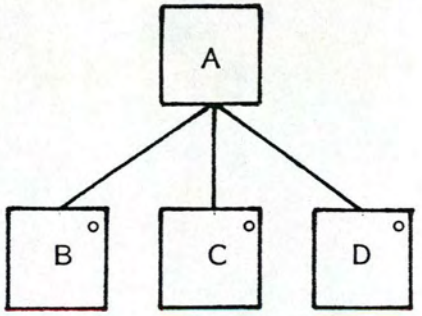
par Eddy POULLET

1984-1987

NOTATIONS UTILISEES

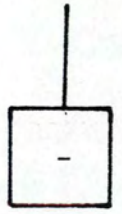


A est un composant séquentiel



A est un composant sélectif

A est un composant itératif



(-) est un composant nul

```
A seq
   B;
   C;
   D;
A end
```

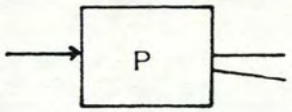
A is a
sequence
component

```
A sel (cond-B)
   B;
A alt (cond-C)
   G;
A alt (cond-D)
   D;
A end
```

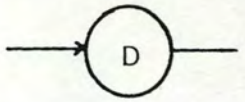
A is a
selection
component

```
A itr while (cond-B)
   B;
A end
```

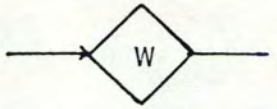
A is an
iteration
component



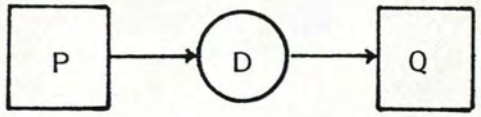
P est un processus



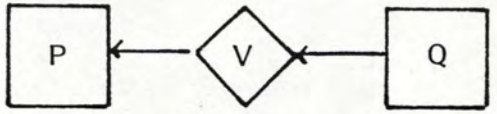
D est une connection par "data-stream"



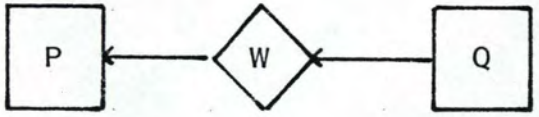
W est une connection par vecteur d'état



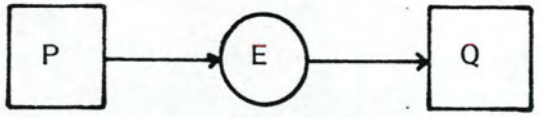
Un P connecté à un Q par un flux de données D écrit par P et lu par Q



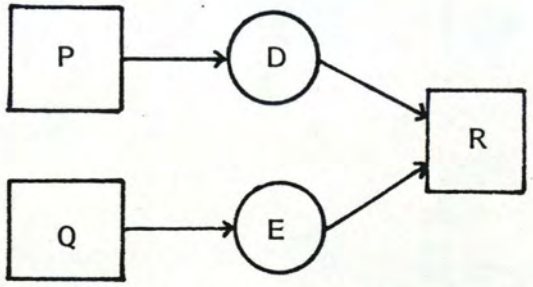
Un P connecté par un Q directement inspectant le vecteur d'état de Q



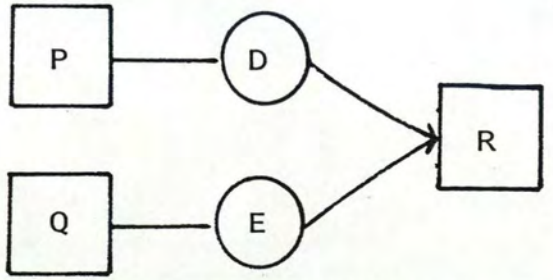
Un P connecté à beaucoup de Q par P examinant directement les vecteurs d'état des Qs



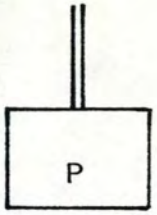
Plusieurs P connectés à plusieurs Q par un flux de données E écrit par Ps et lu par les Qs



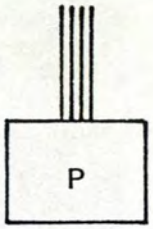
R mélange ses entrées de données D et E par un "merge" fixe ou influencé par la valeur des données



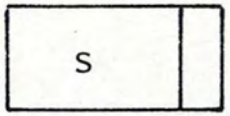
R mélange ses entrées de données D et E par un "rough merge"



P est un processus inverti avec respect à un de ses "data-stream"

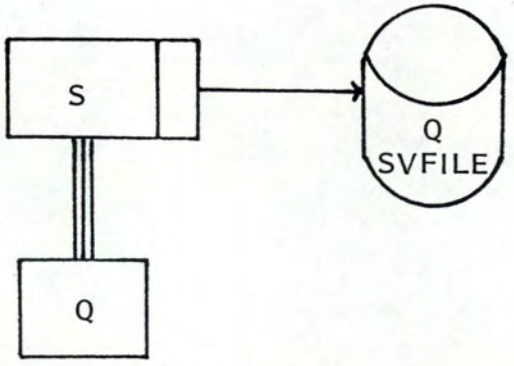


P est un processus inverti avec respect à trois de ses data-stream

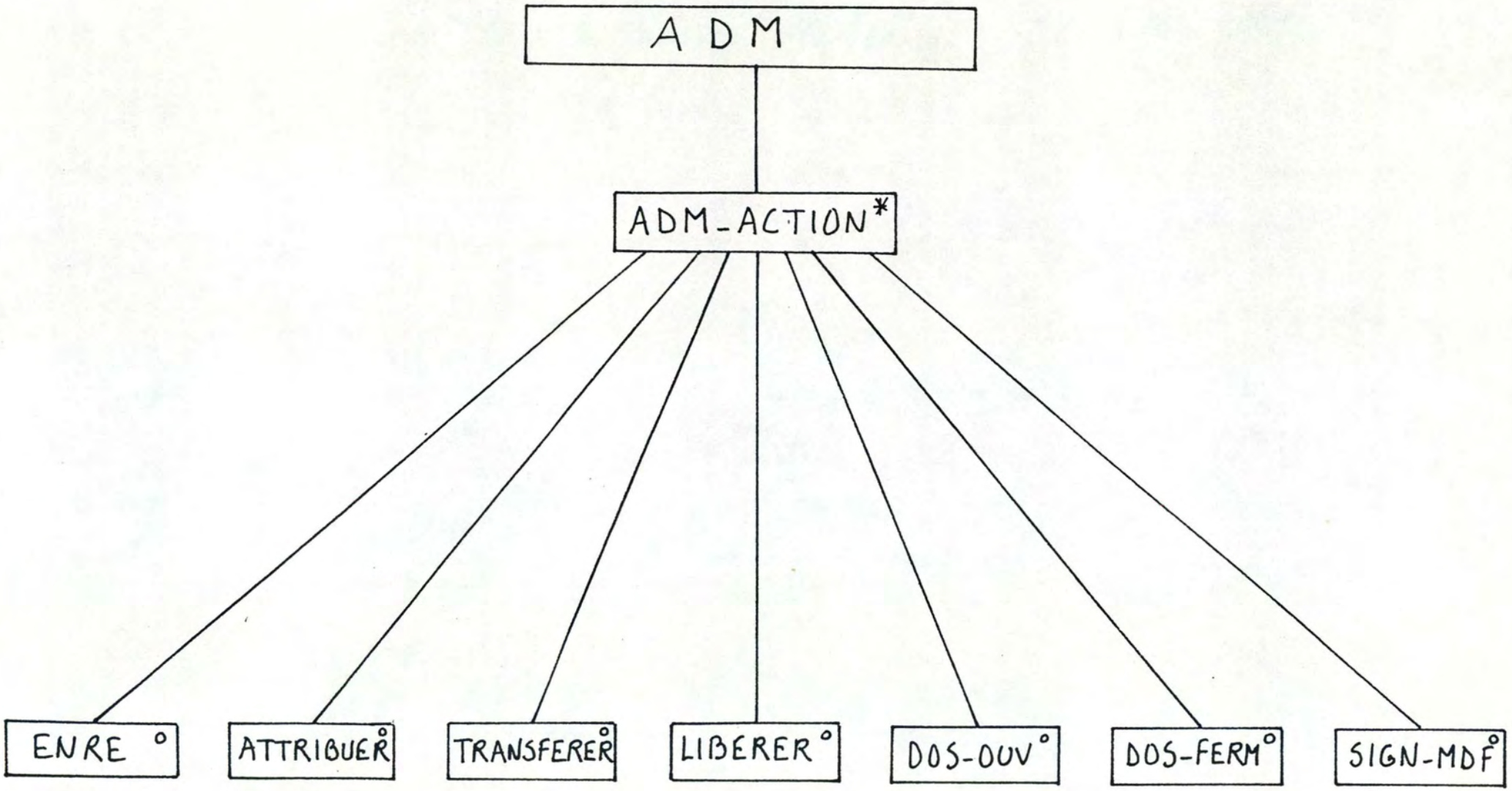


S est un processus séquenceur non présent dans le système de spécification, introduit dans l'étape d'implémentation

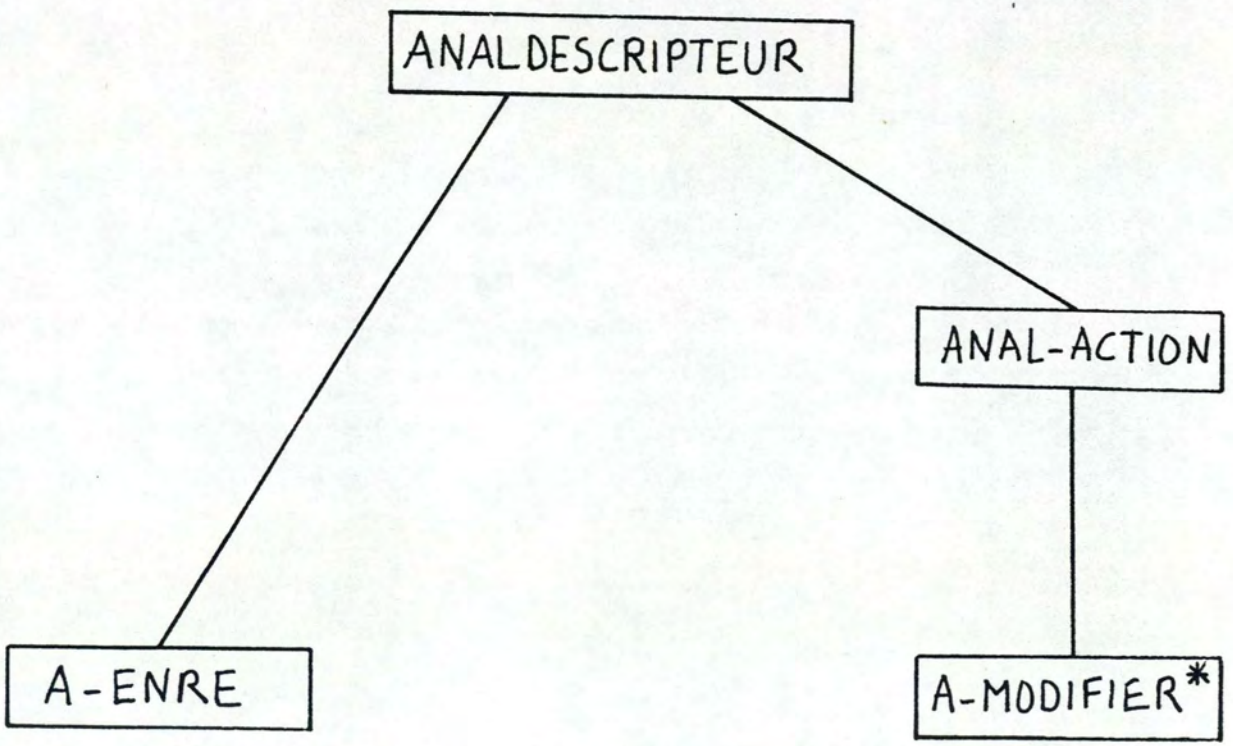
ABC-SVFILE est un facteur (à accès direct) de tous les vecteurs d'état des processus ABC, qui ont été séparés du programme d'ABC

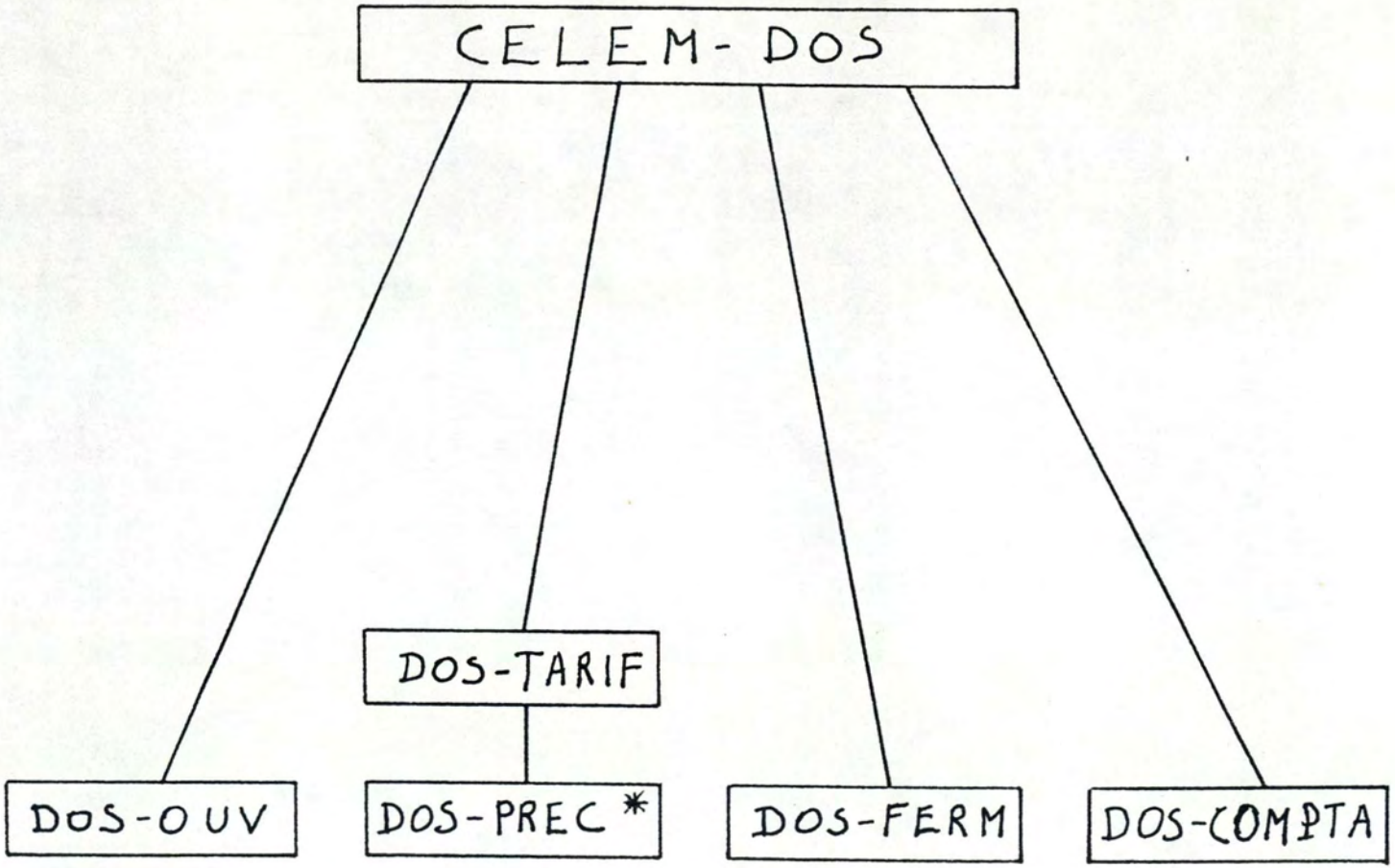


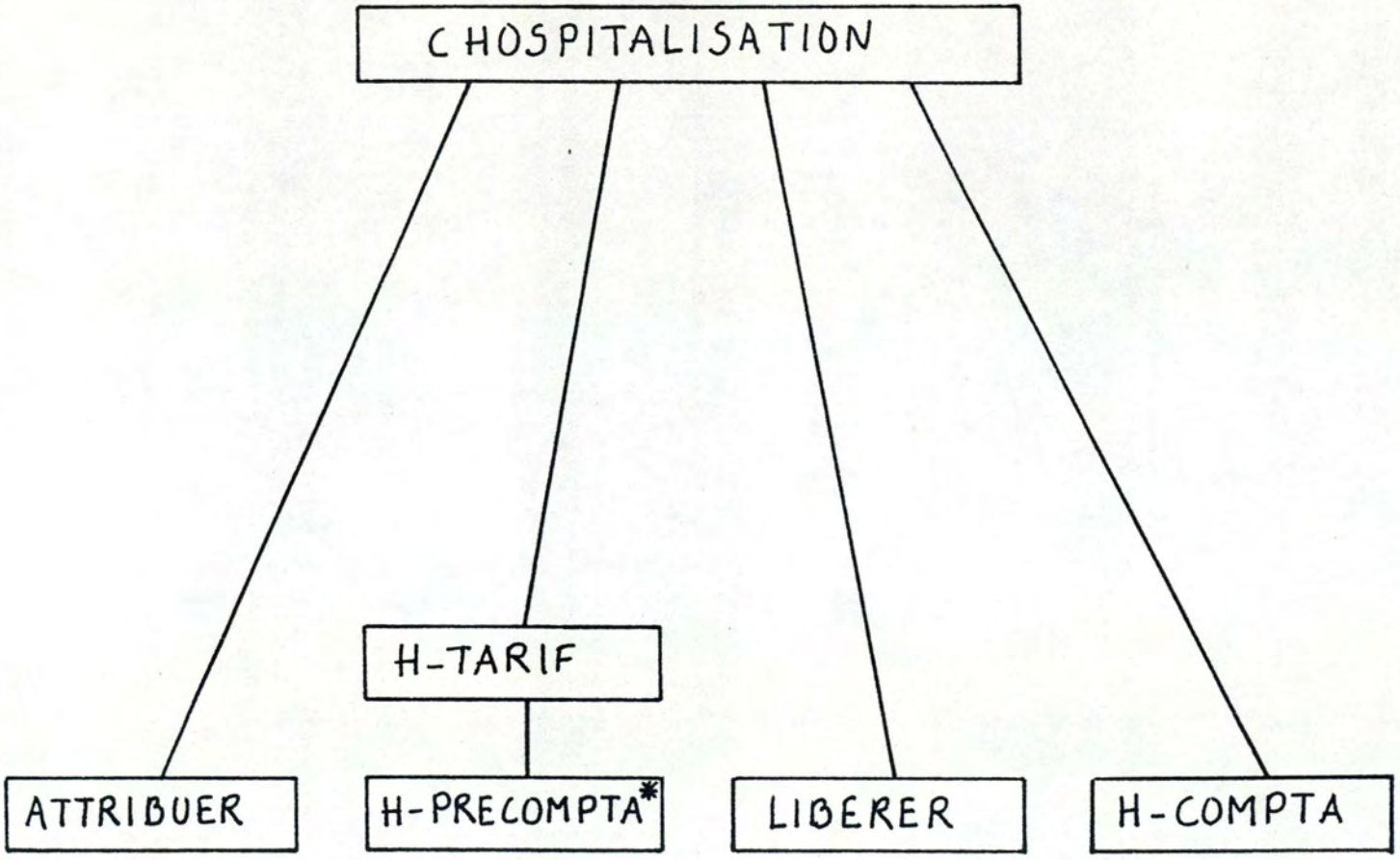
Q est inversé avec respect à deux "data-stream" et ses vecteurs d'état ont été séparés. Le processus séquenceur S retire et stocke le vecteur d'état de Q

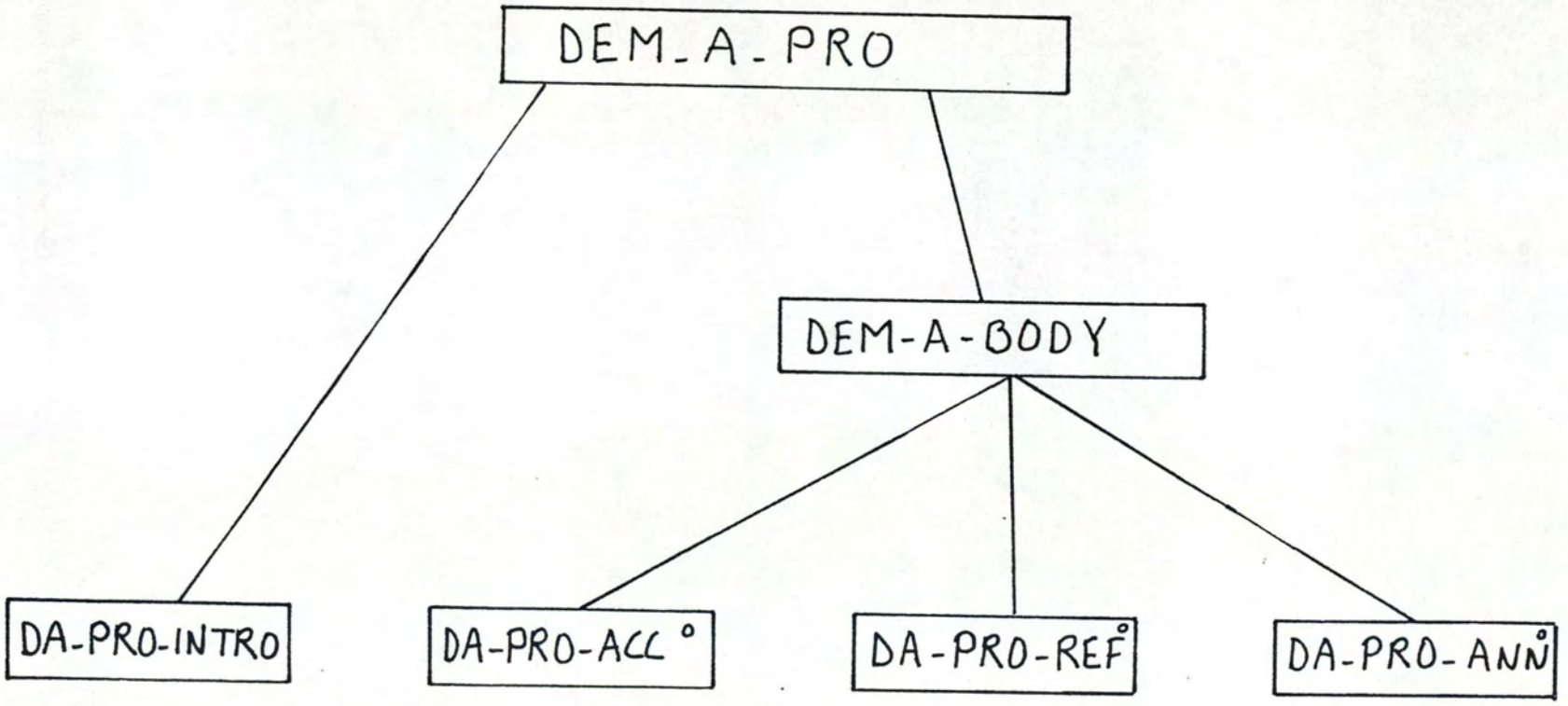


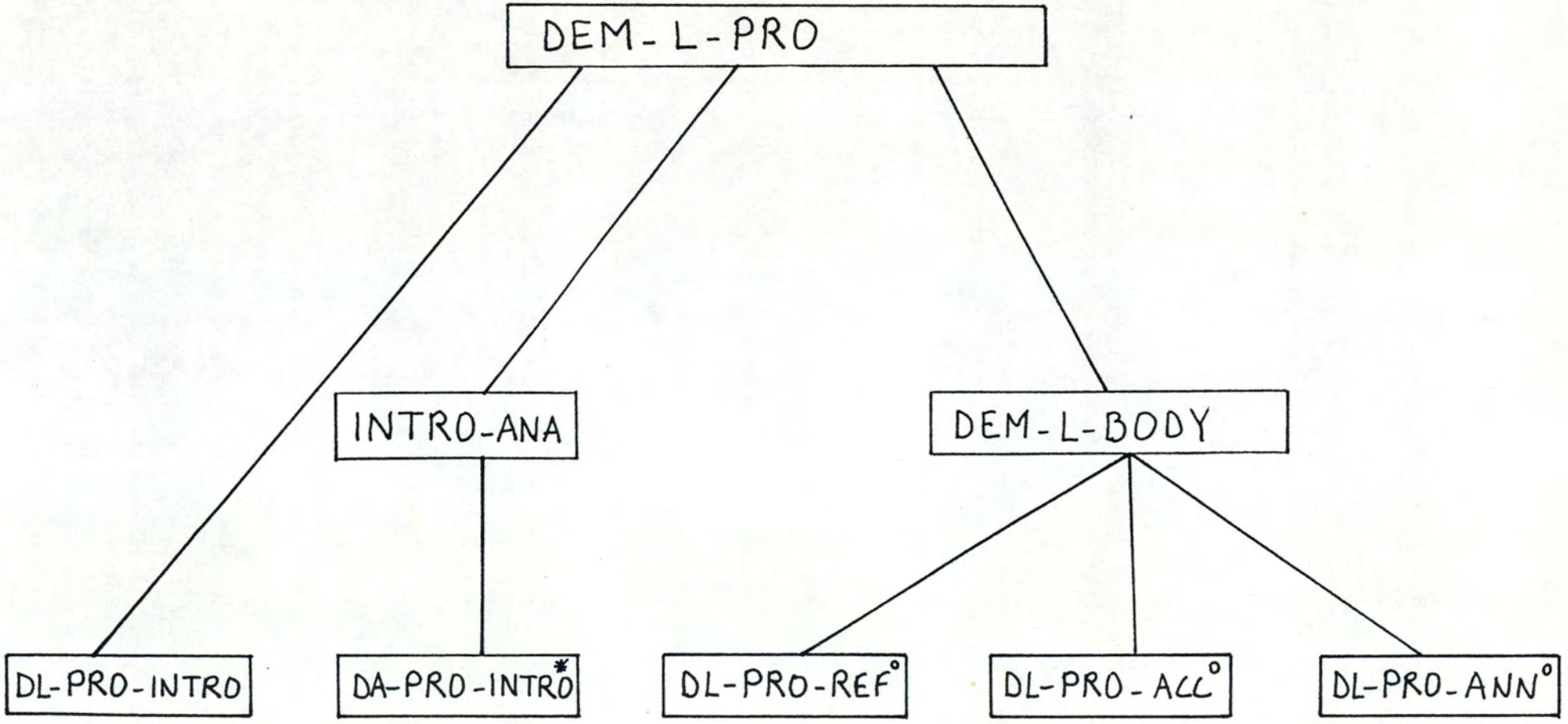
ENTITY STRUCTURE DIAGRAM

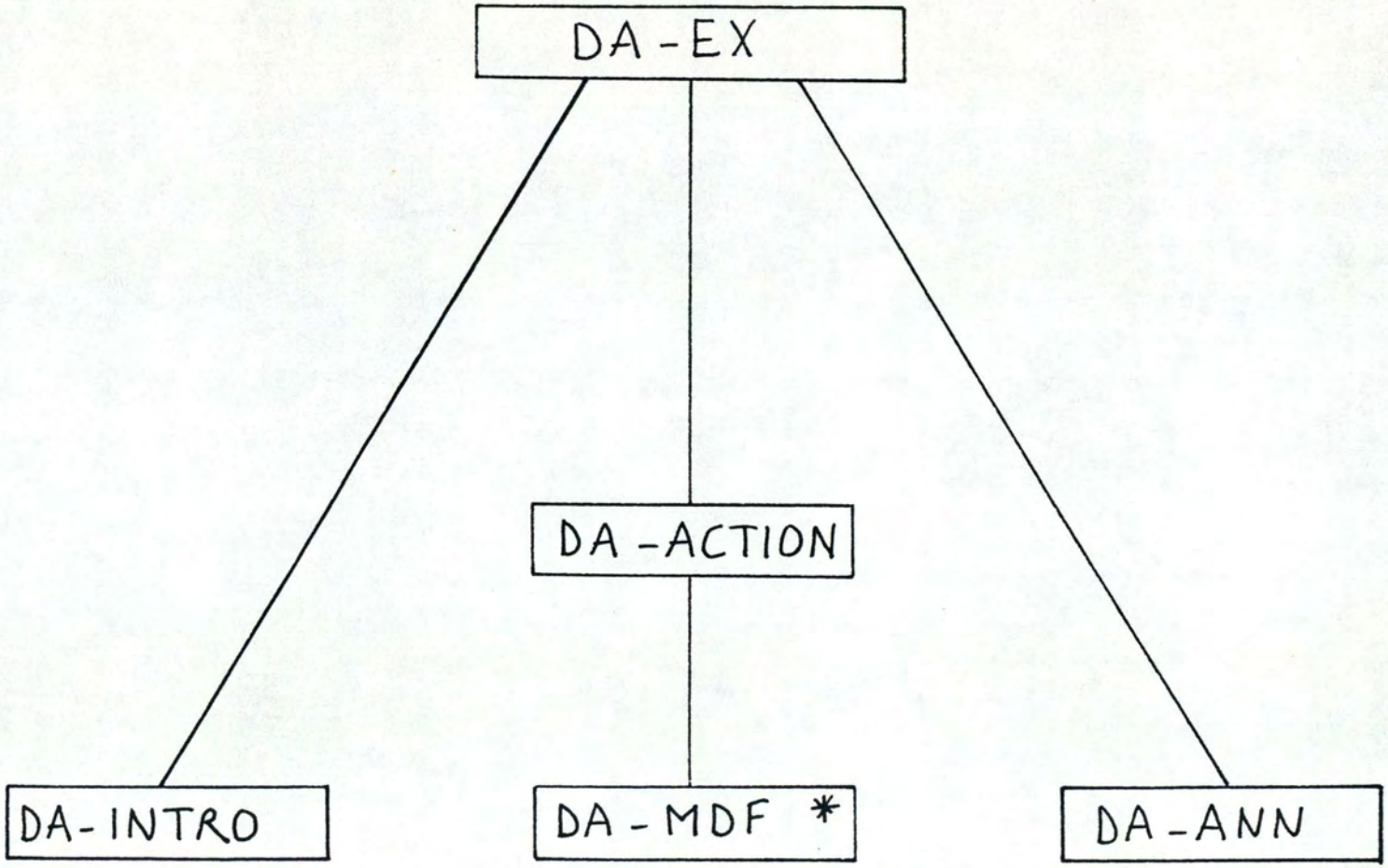

ANALDESCRIPTEUR
ANAL-ACTION
A-ENRE
A-MODIFIER*

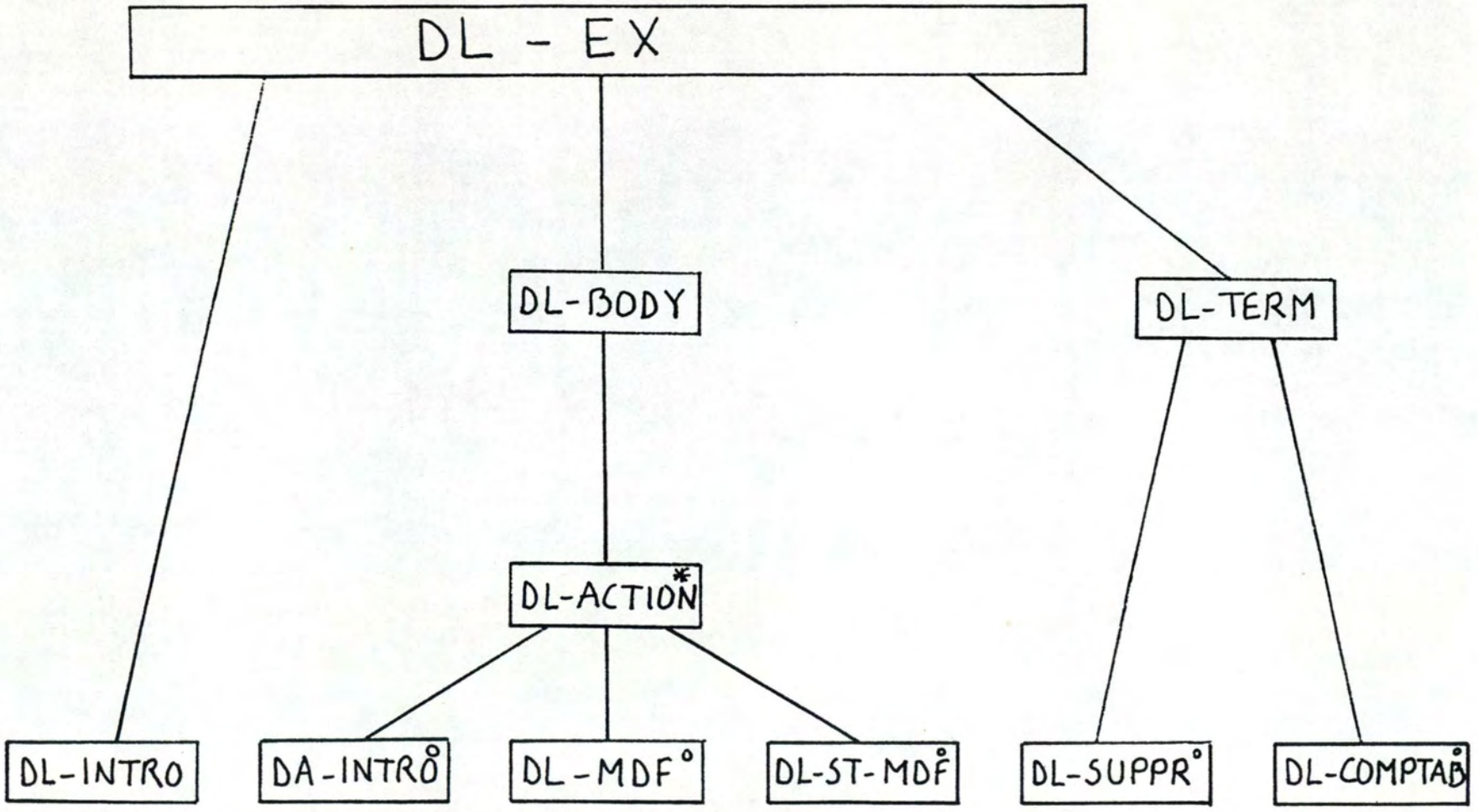

CELEM-DOS
DOS-OUV
DOS-TARIF
DOS-PREC *
DOS-FERM
DOS-COMPTA

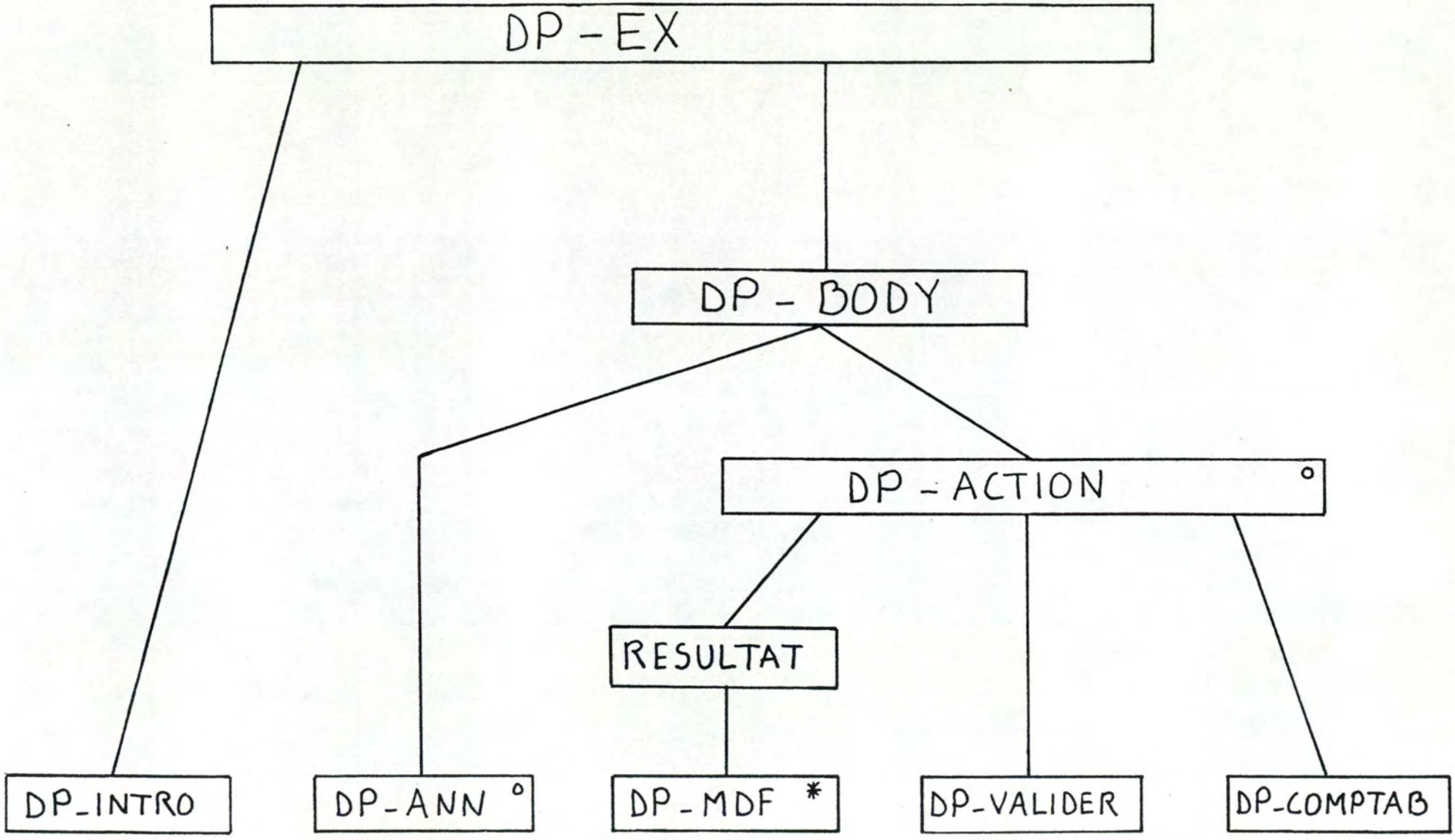

C HOSPITALISATION
H-TARIF
ATTRIBUER
H-PRECOMPTA*
LIBERER
H-COMPTA

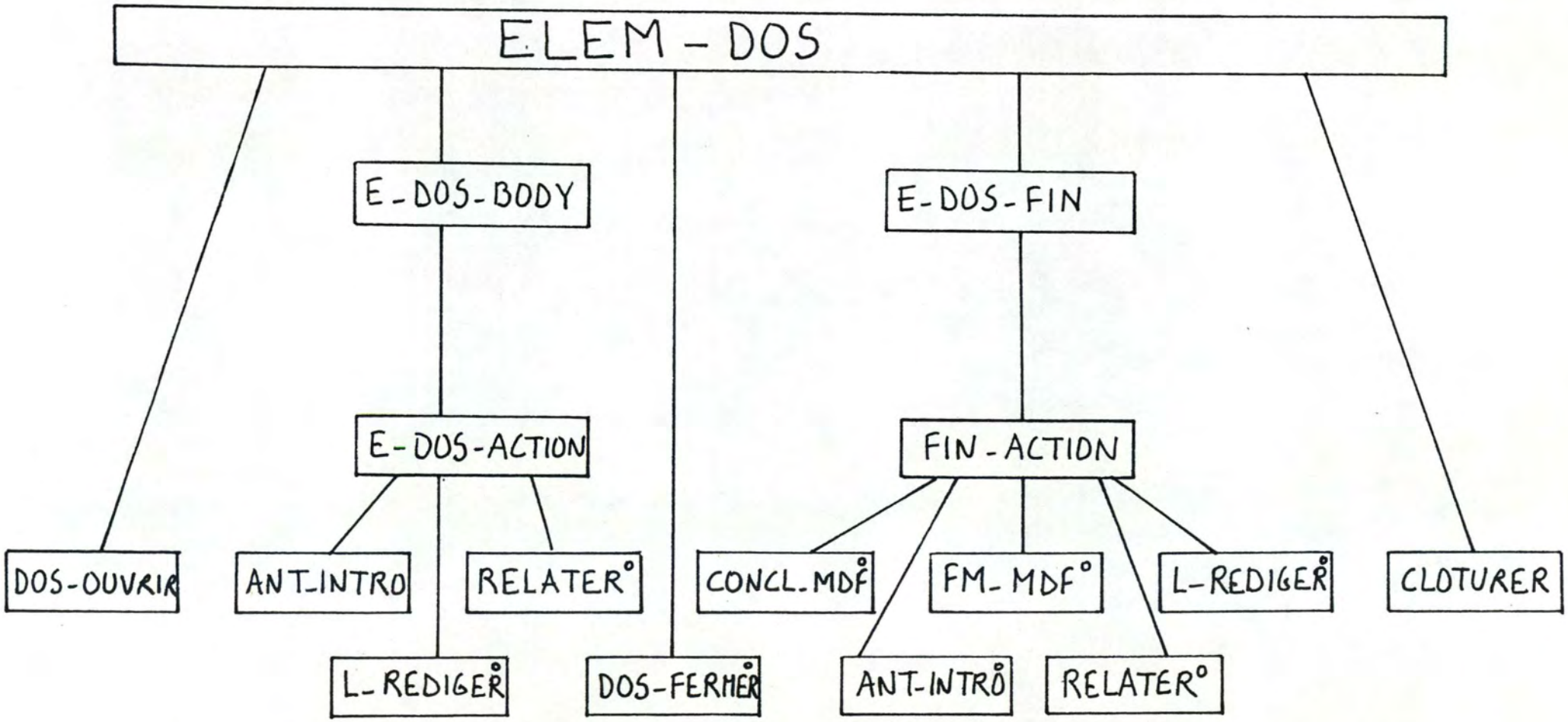

DEM_A_PRO
DEM-A-BODY
DA-PRO-INTRO
DA-PRO-ACC°
DA-PRO-REF°
DA-PRO-ANN°

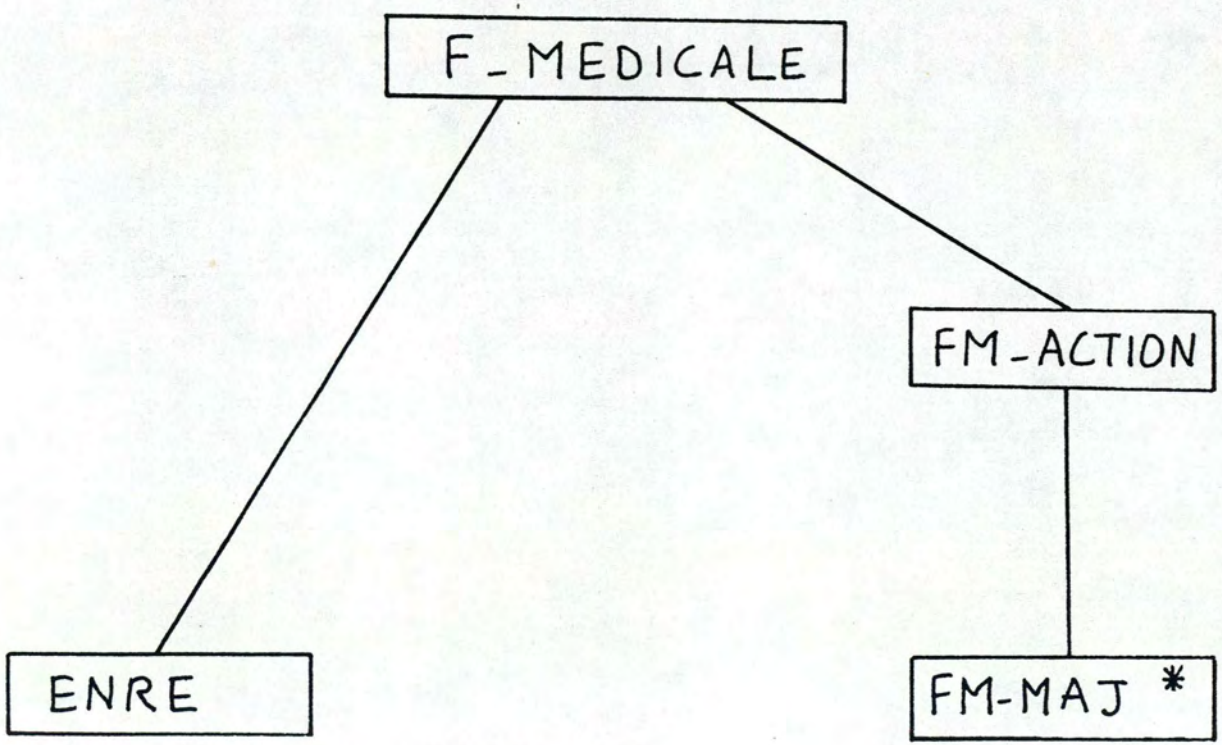

DEM-L-PRO
INTRO-ANA
DEM-L-BODY
DL-PRO-INTRO
DA-PRO-INTRO*
DL-PRO-REF°
DL-PRO-ACC°
DL-PRO-ANN°

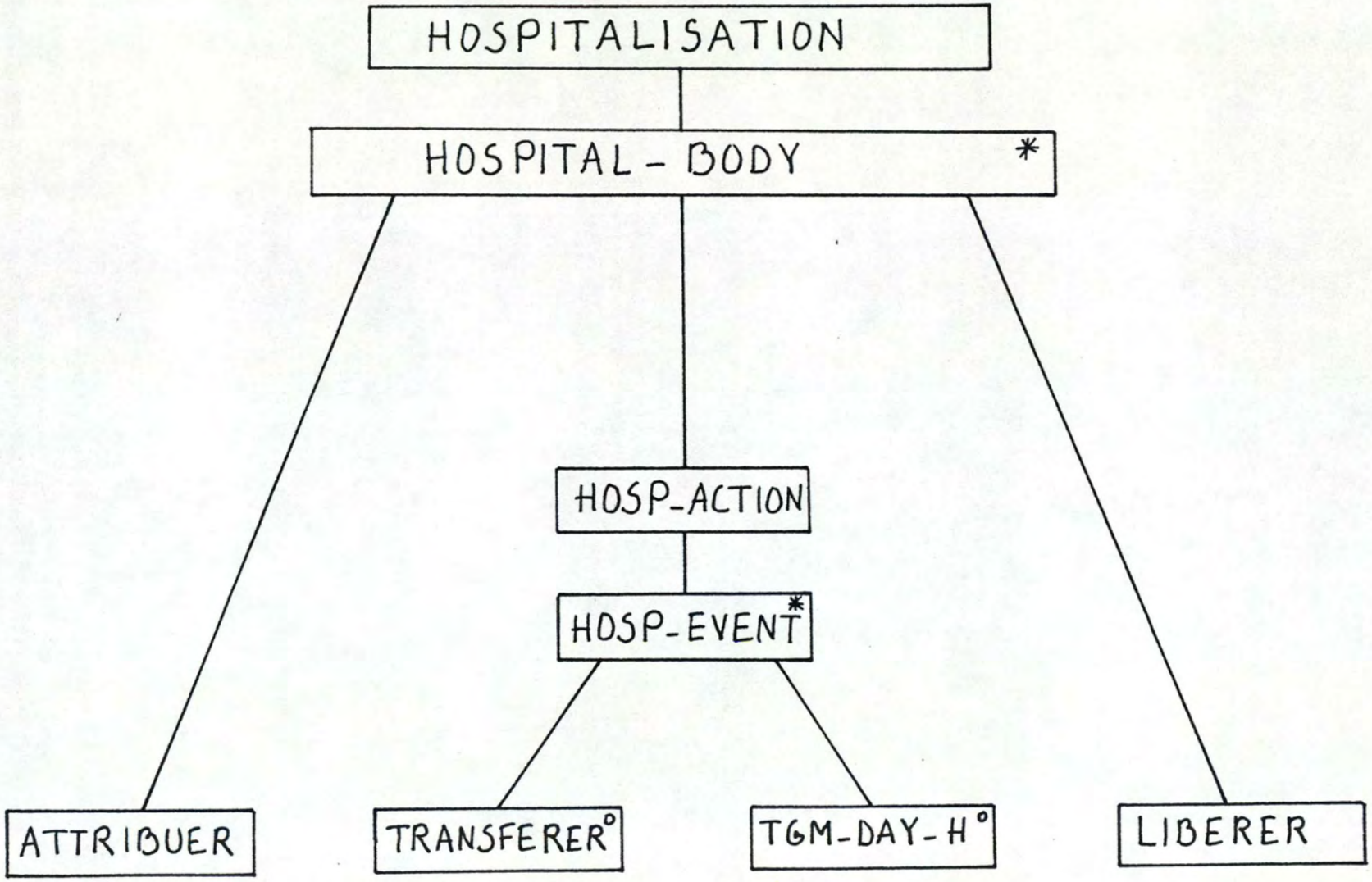

DA-EX
DA-ACTION
DA-INTRO
DA-MDF *
DA-ANN

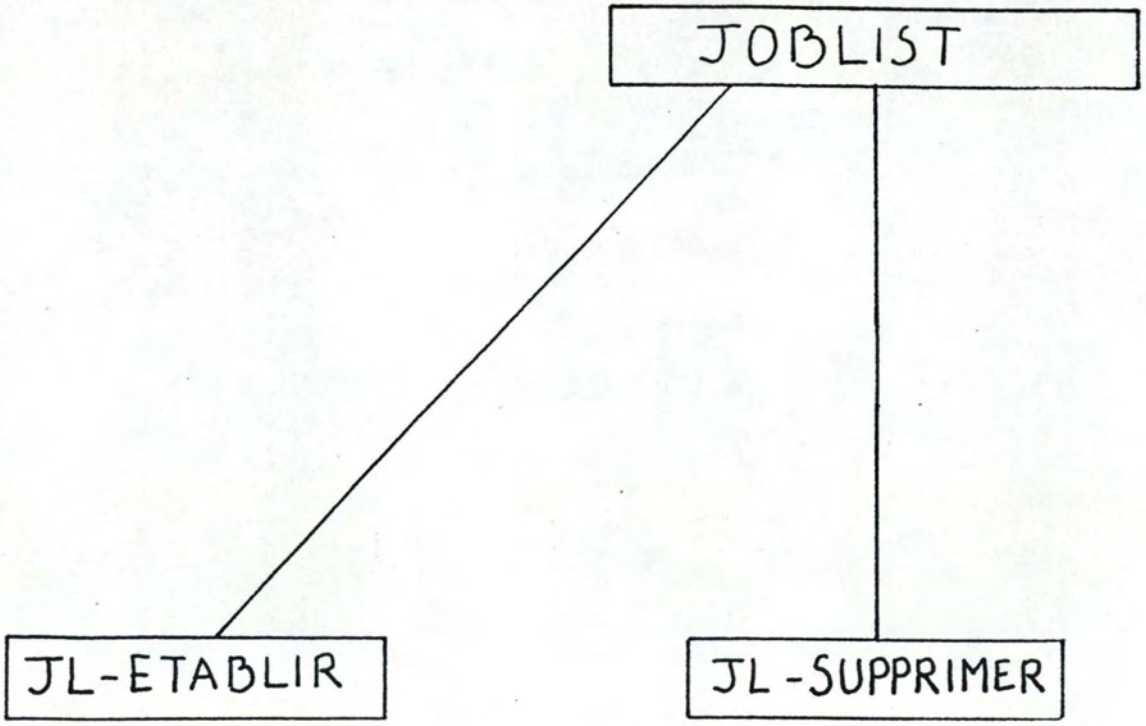

DL - EX
DL-BODY
DL-TERM
DL-ACTION*
DL-INTRO
DA-INTRO°
DL-MDF°
DL-ST-MDF
DL-SUPPR°
DL-COMPTAB°

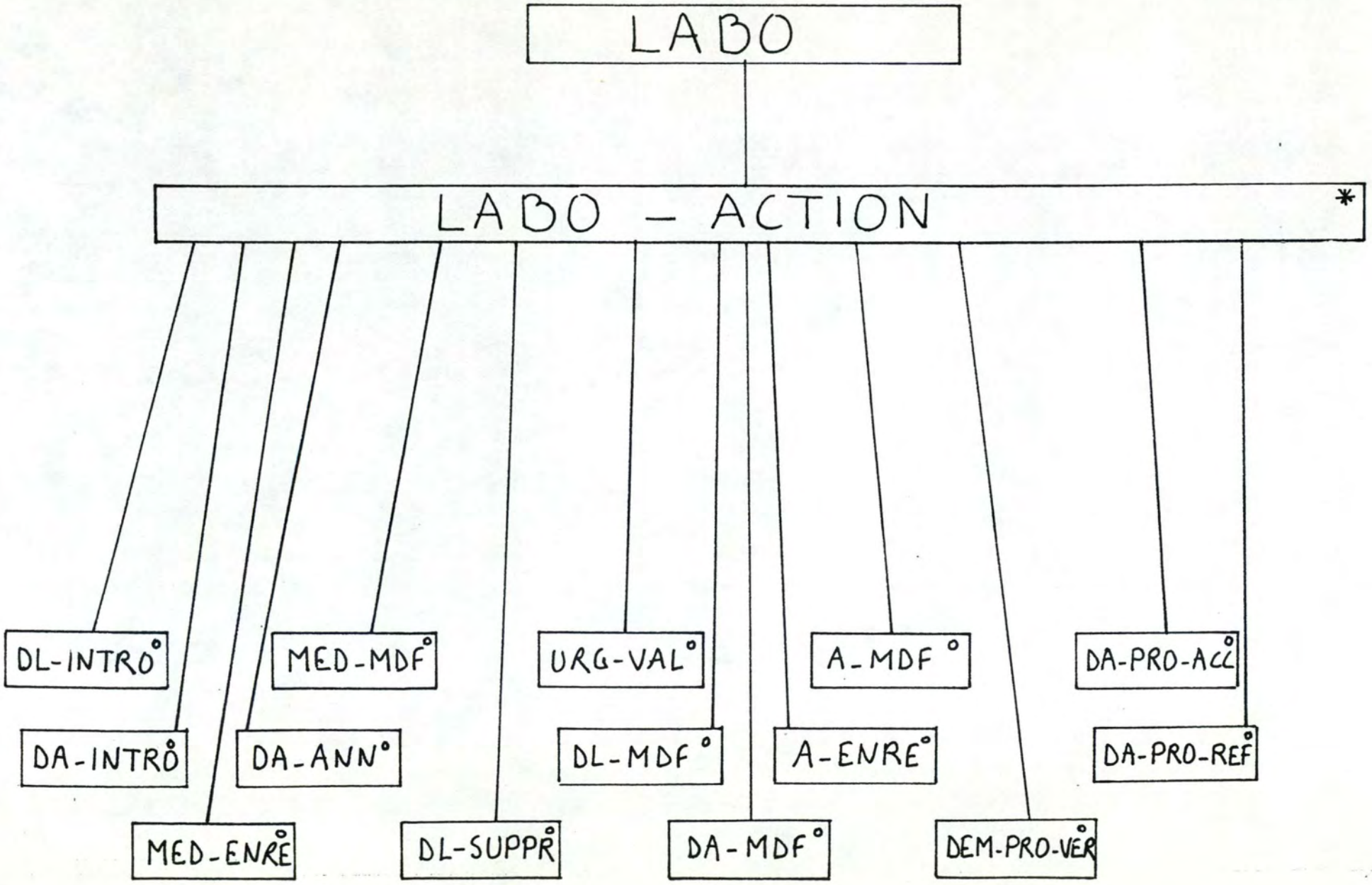

DP - EX
DP - BODY
DP - ACTION °
RESULTAT
DP_INTRO
DP - ANN °
DP - MDF *
DP_VALIDER
DP_COMPTAB

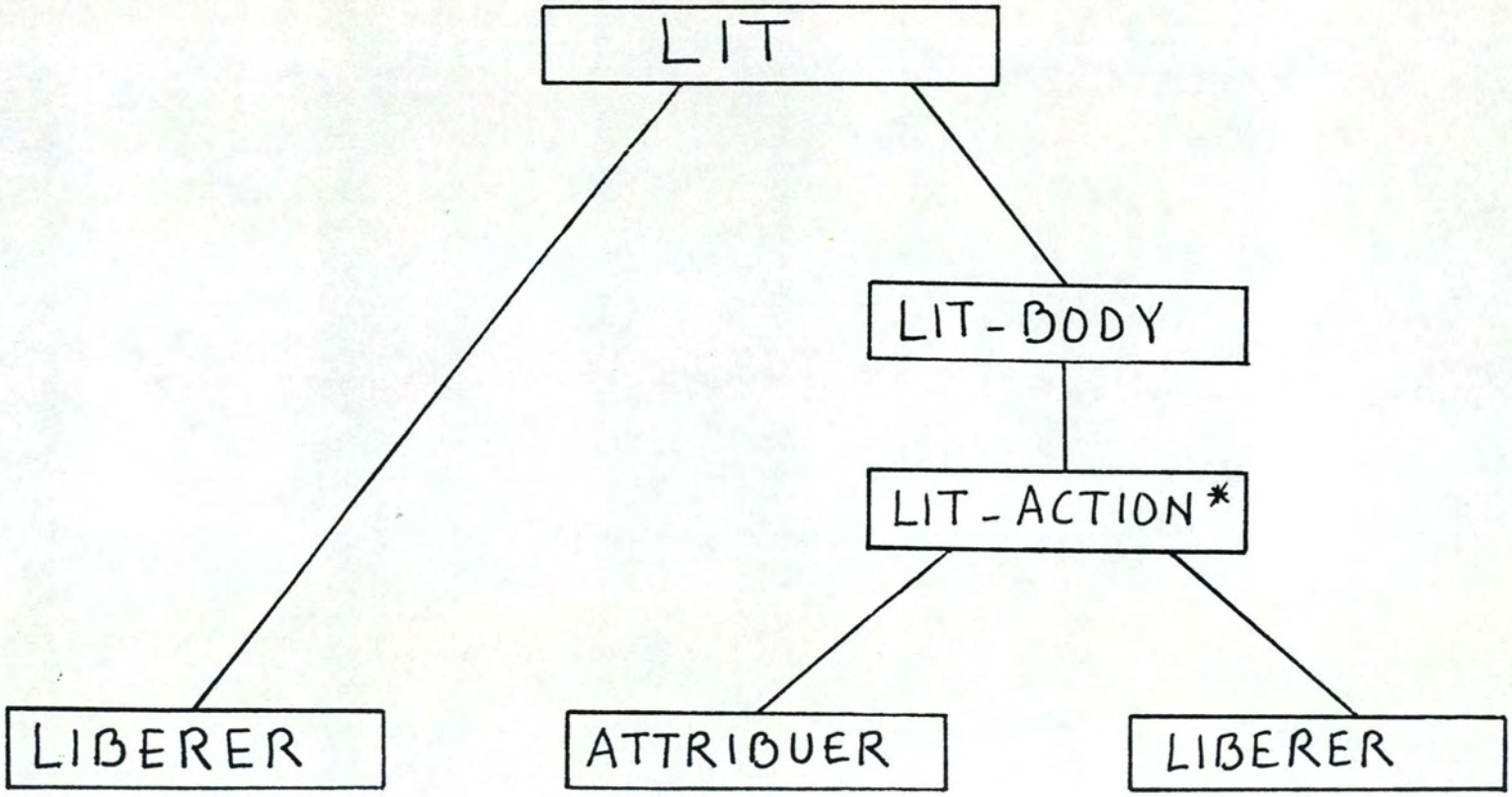

ELEM - DOS
E-DOS-BODY
E-DOS-FIN
E-DOS-ACTION
FIN-ACTION
DOS-OUVRIR
ANT_INTRO
RELATER°
CONCL_MDF
FM_MDF°
L-REDIGER
CLOTURER
L-REDIGER
DOS-FERMER
ANT-INTRO
RELATER°

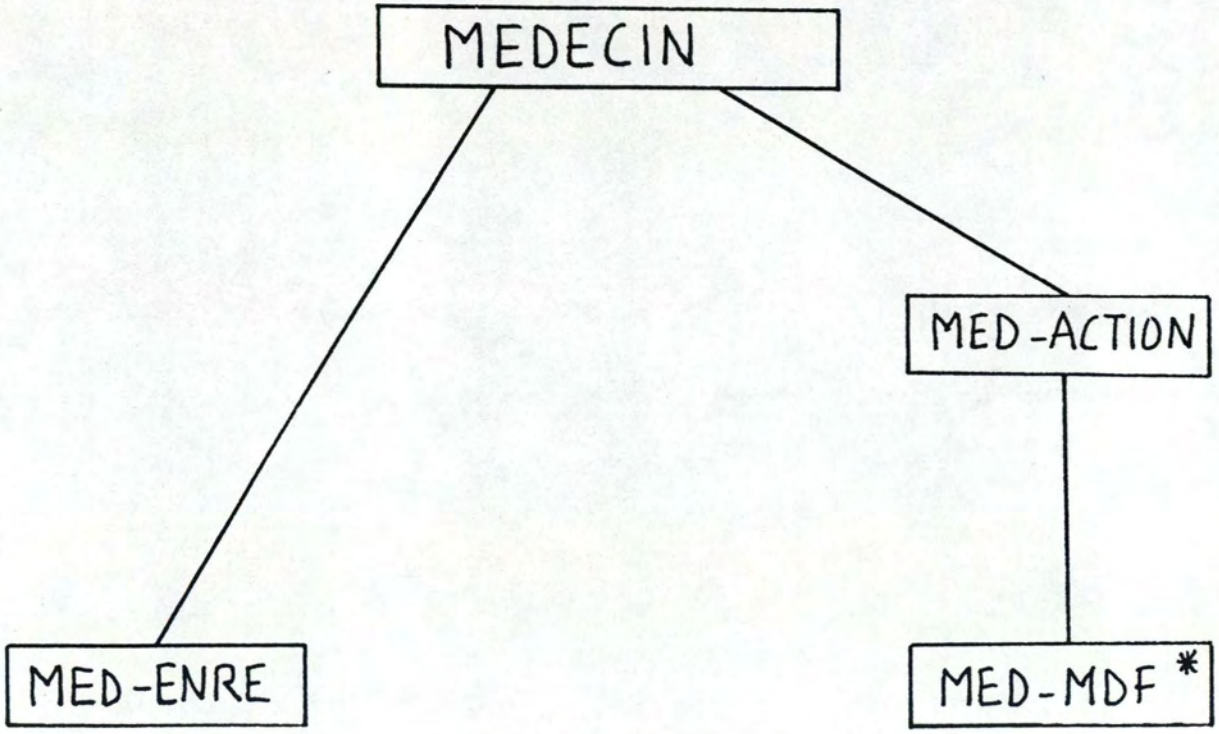

F_MEDICALE
FM-ACTION
ENRE
FM-MAJ *

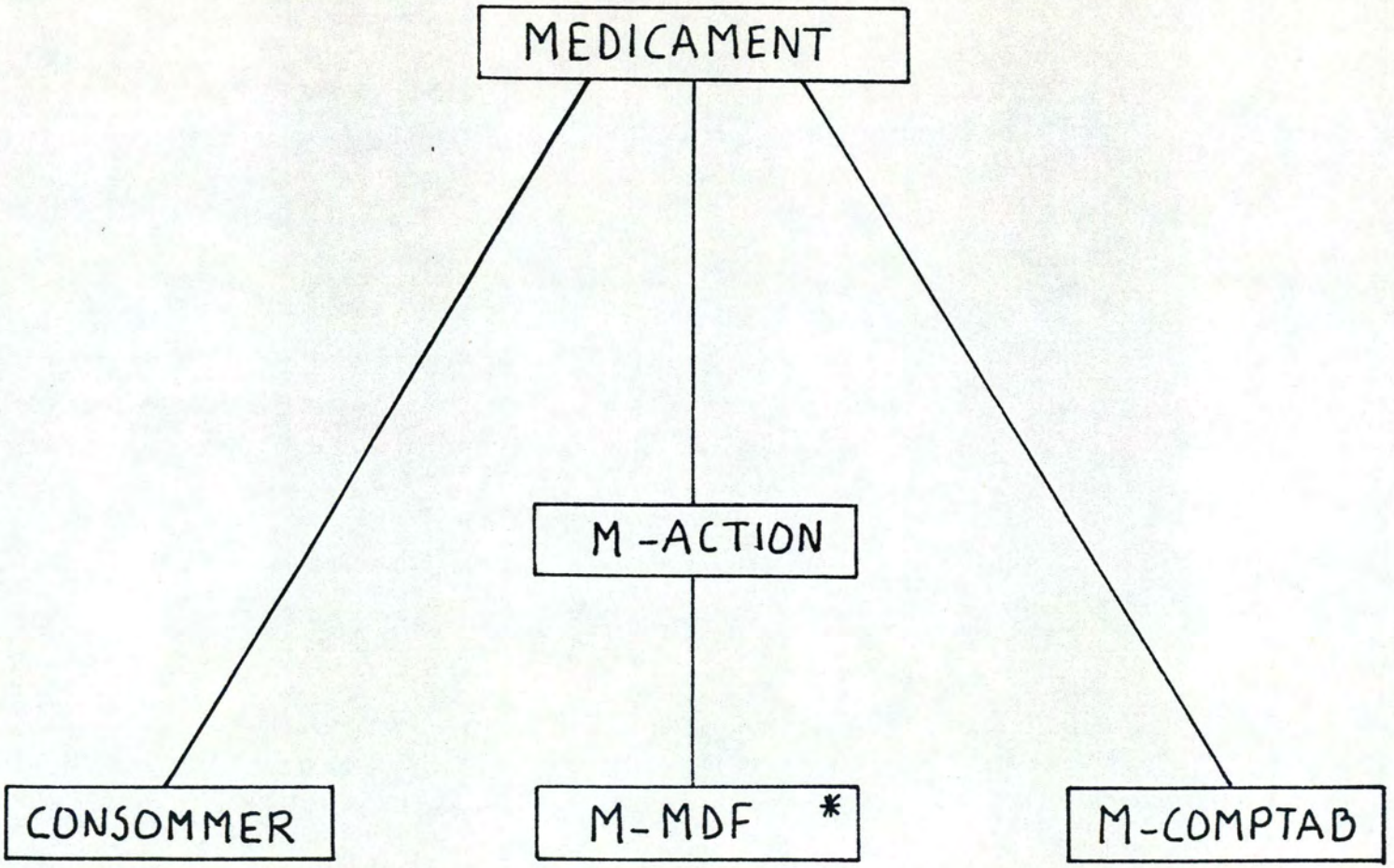

HOSPITALISATION
HOSPITAL - BODY *
HOSP_ACTION
HOSP-EVENT *
ATTRIBUER
TRANSFERER°
TGM-DAY-H°
LIBERER

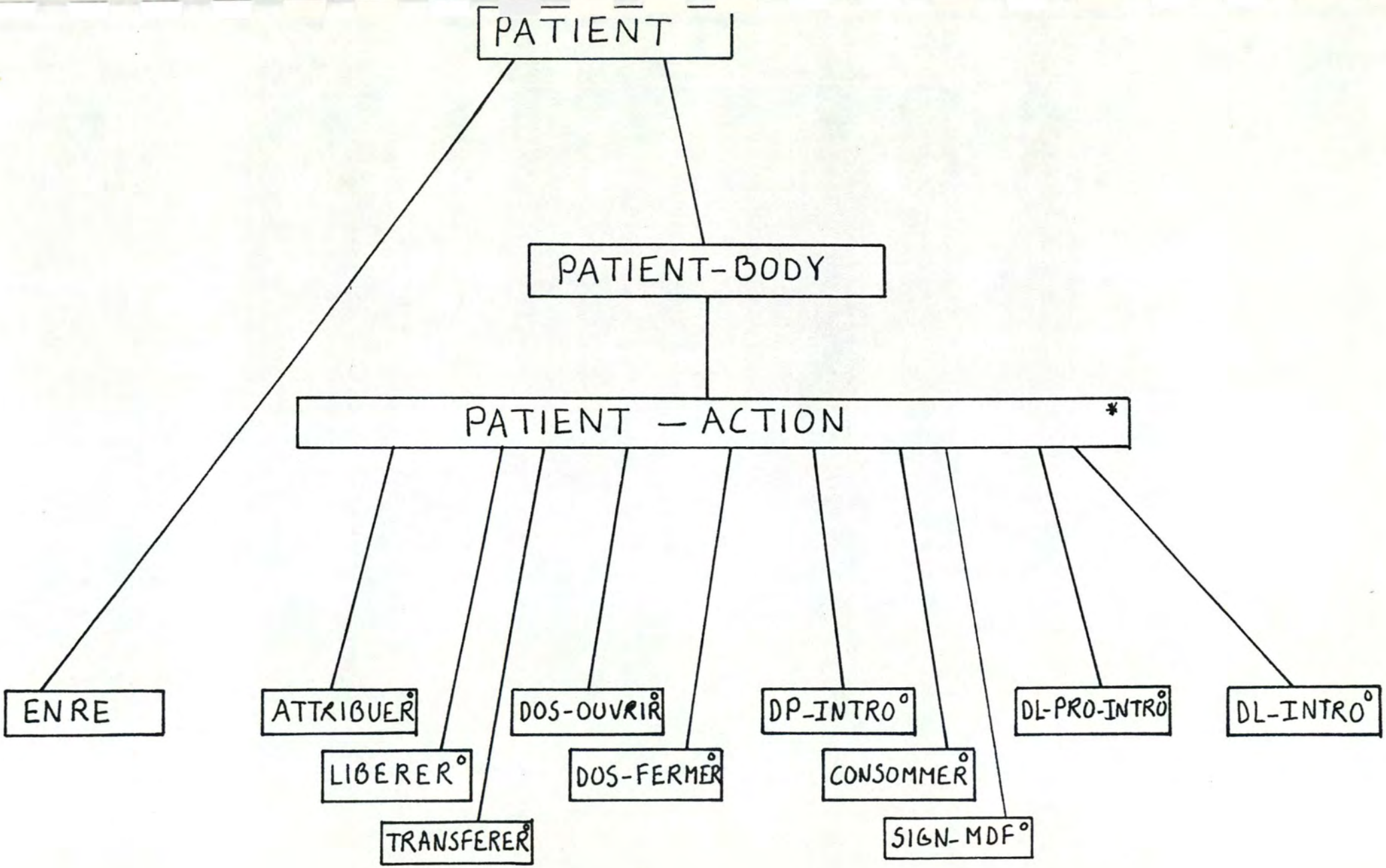

JOBLIST
JL-ETABLIR
JL-SUPPRIMER

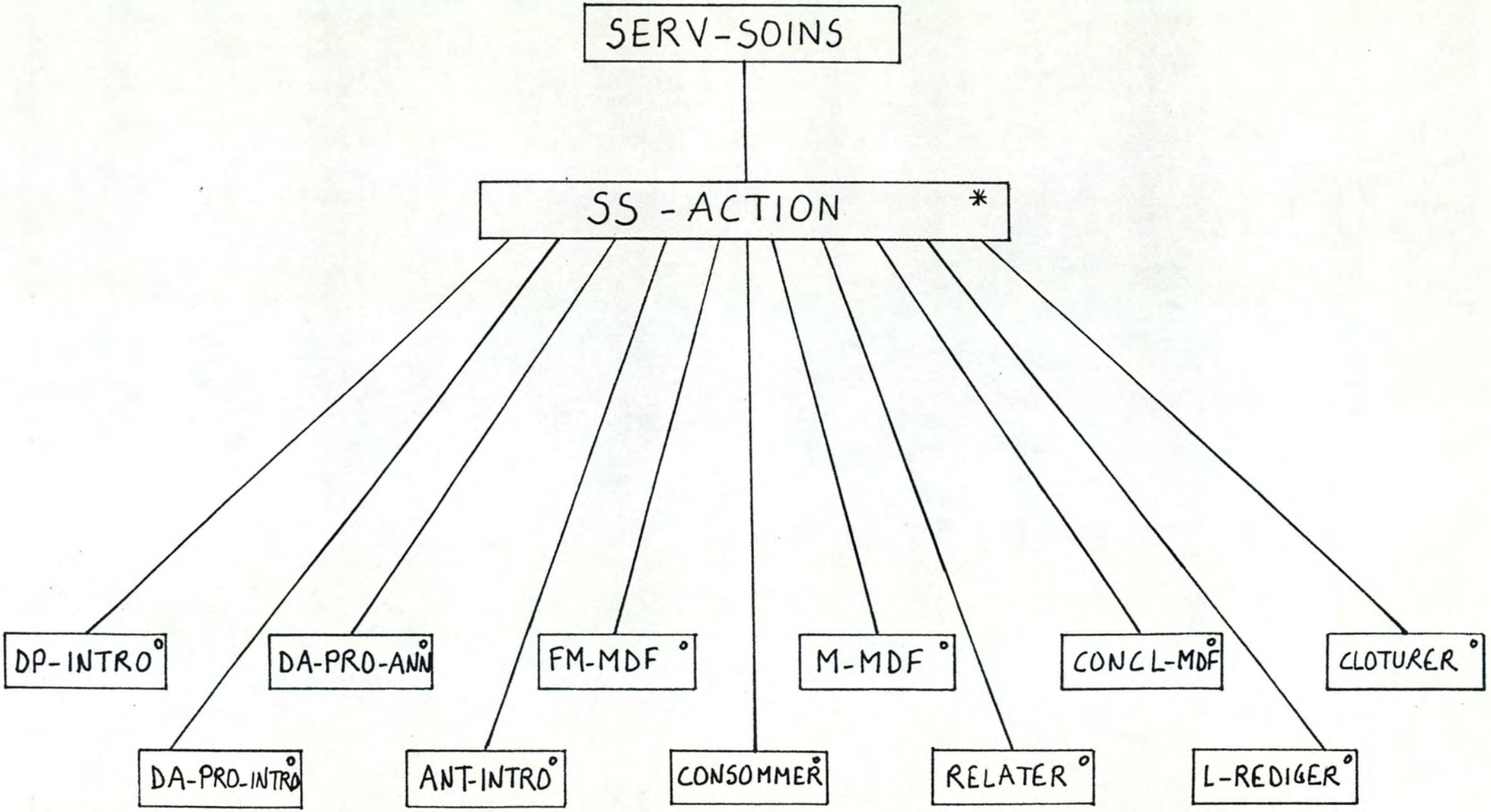

LABO
LABO - ACTION *
DL-INTRO°
DA-INTRÔ
MED-ENRÉ
DA-ANN°
MED-MDF°
DL-SUPPR
DL-MDF°
URG-VAL°
DA-MDF°
A-ENRE°
A-MDF°
DEM-PRO-VÉR
DA-PRO-REF
DA-PRO-ACC°

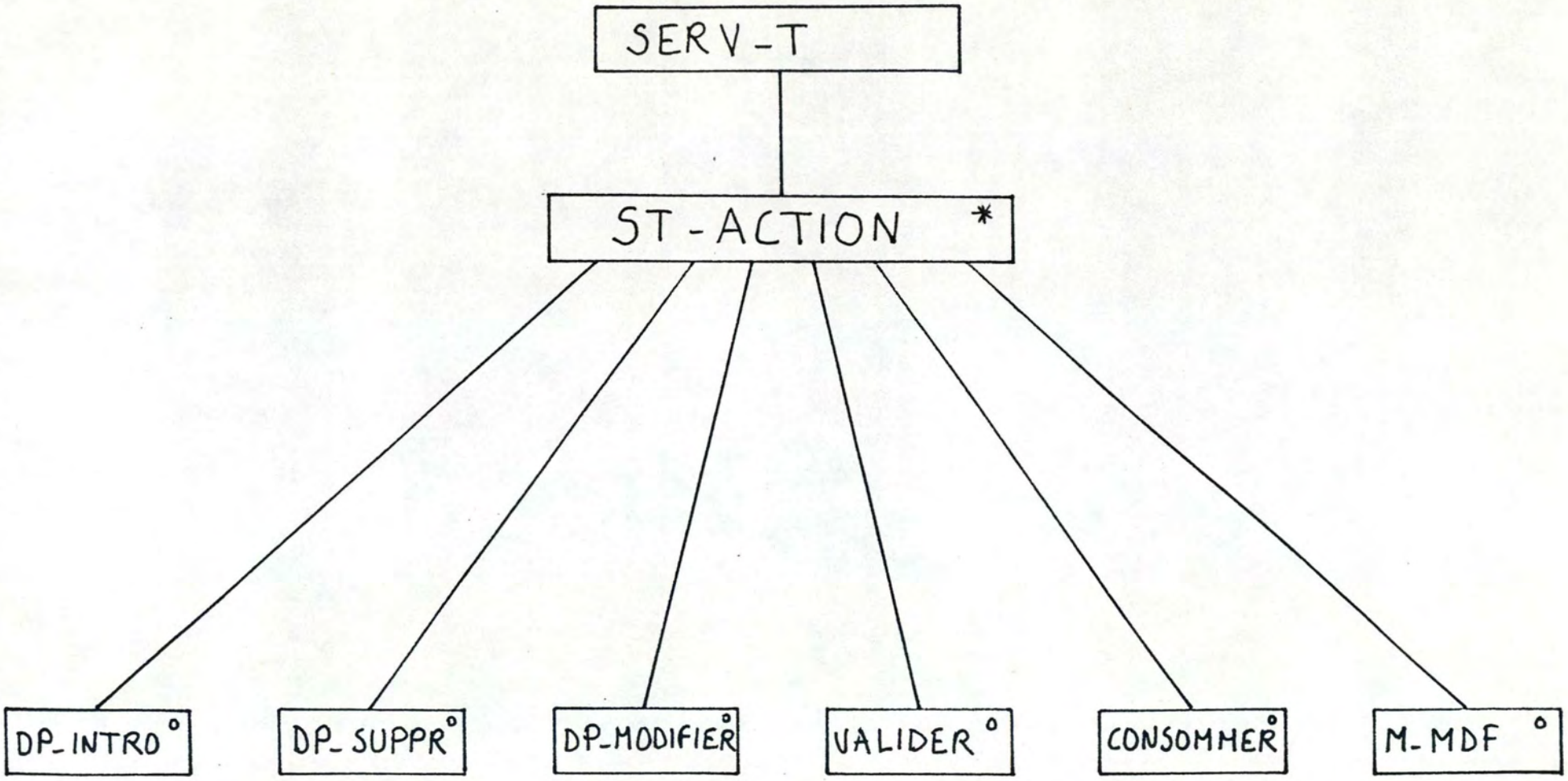

LIT
LIT-BODY
LIT-ACTION*
LIBERER
ATTRIBUER
LIBERER

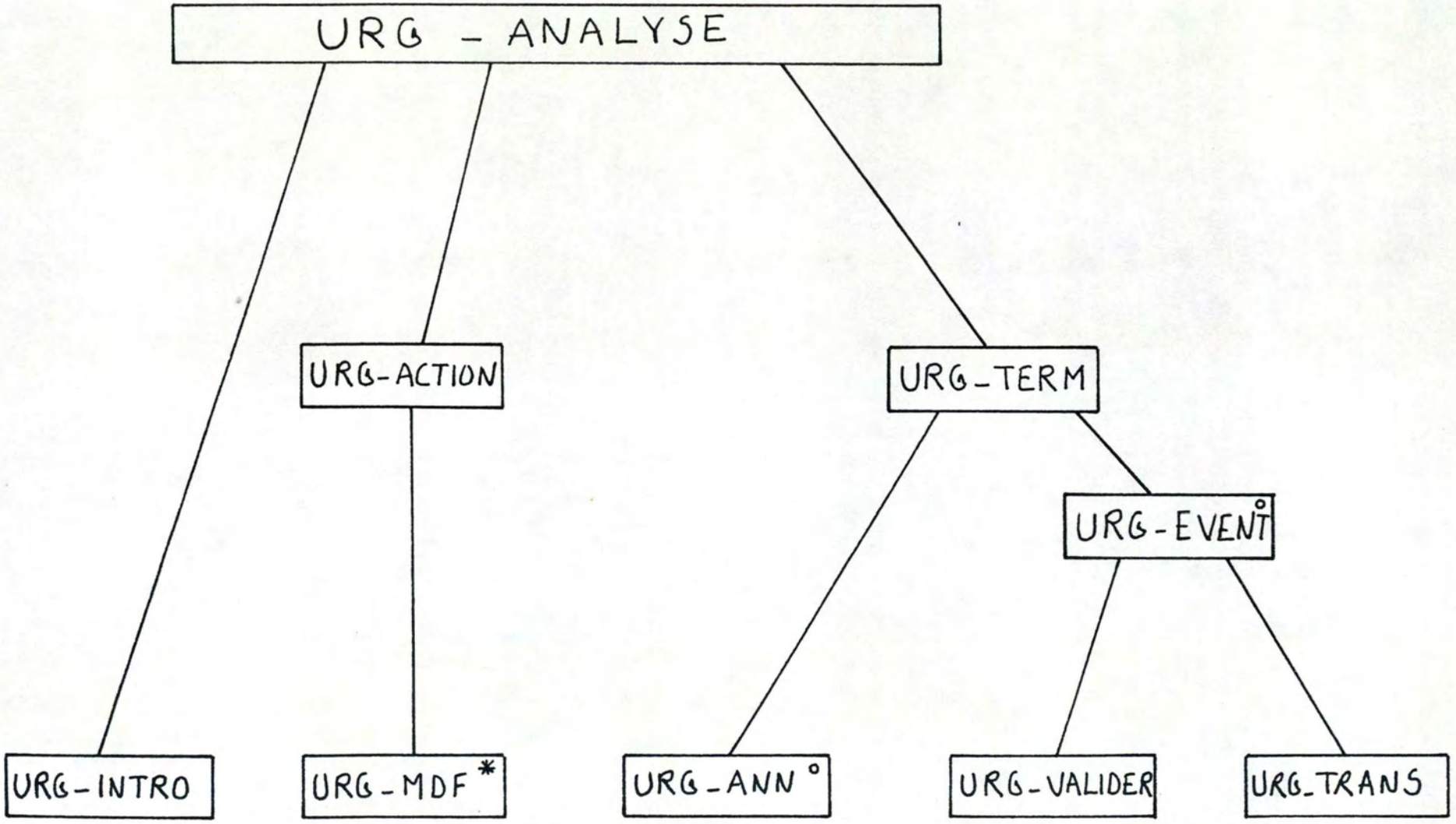

MEDECIN
MED-ACTION
MED-ENRE
MED-MDF *

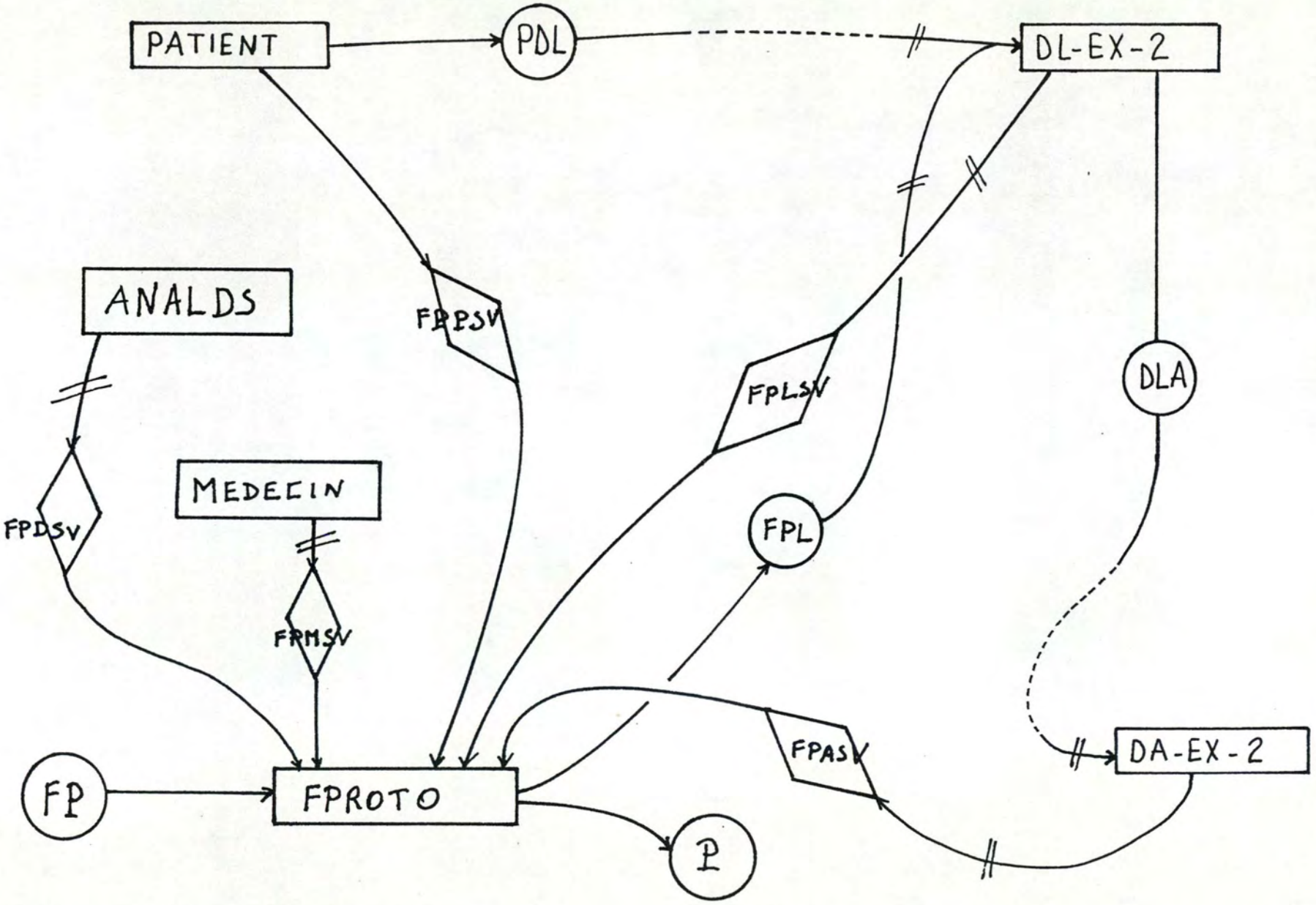

MEDICAMENT
M-ACTION
CONSOMMER
M-MDF *
M-COMPTAB


PATIENT
PATIENT-BODY
PATIENT – ACTION *
ENRE
ATTRIBUER°
LIBERER°
TRANSFERER°
DOS-OUVRIR°
DOS-FERMER°
DP_INTRO°
CONSOMMER°
SIGN_MDF°
DL-PRO-INTRO°
DL_INTRO°

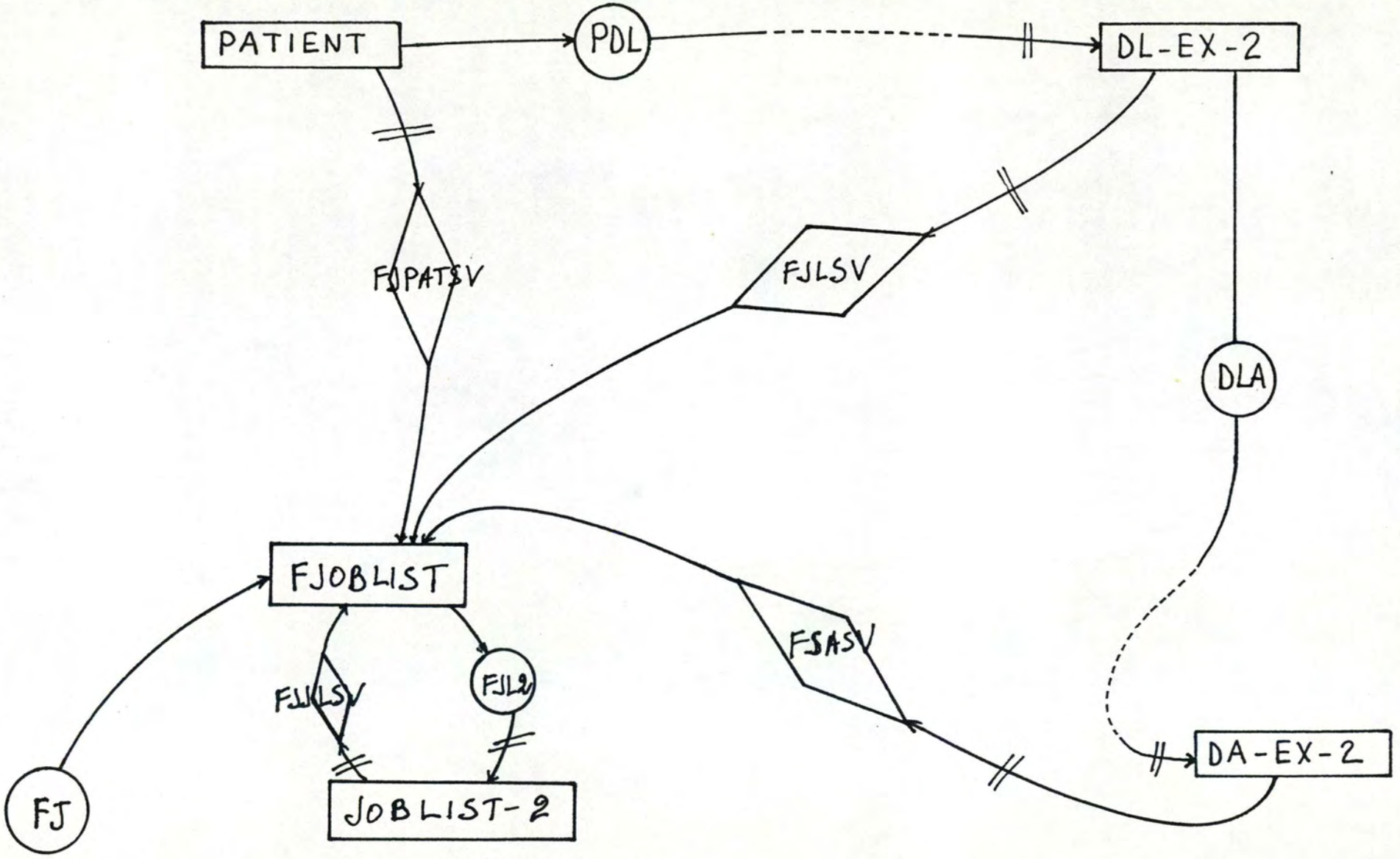

SERV-SOINS
SS - ACTION *
DP-INTRO°
DA-PRO-ANN°
FM-MDF °
M-MDF °
CONCL-MDF°
CLOTURER °
DA-PRO-INTRO°
ANT-INTRO°
CONSOMMER°
RELATER °
L-REDIGER°


SERV-T
ST-ACTION *
DP_INTRO °
DP_SUPPR °
DP_MODIFIER °
VALIDER °
CONSOMMER °
M_MDF °

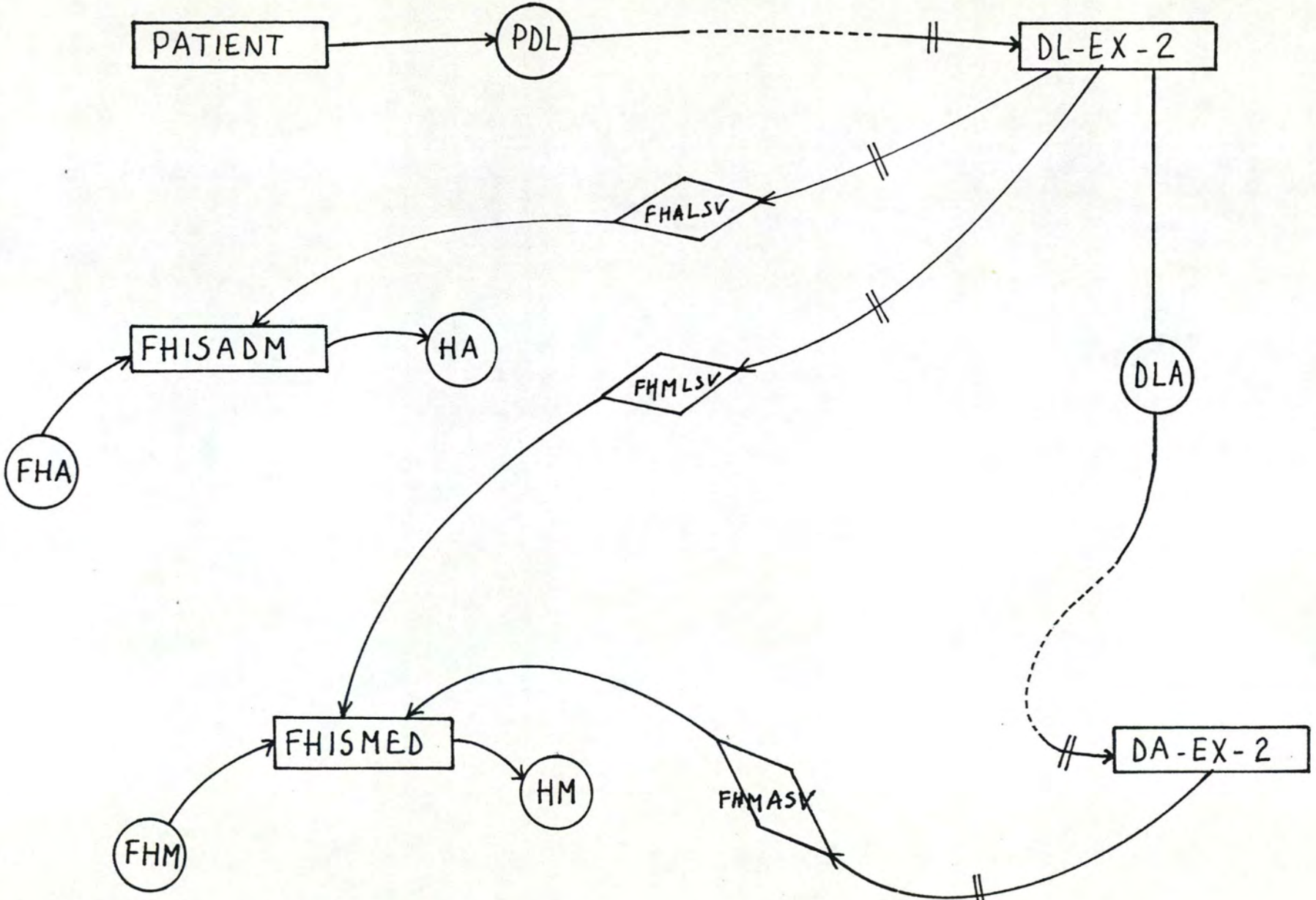

URG - ANALYSE
URG-ACTION
URG_TERM
URG-EVENT°
URG-INTRO
URG-MDF *
URG-ANN °
URG-VALIDER
URG_TRANS



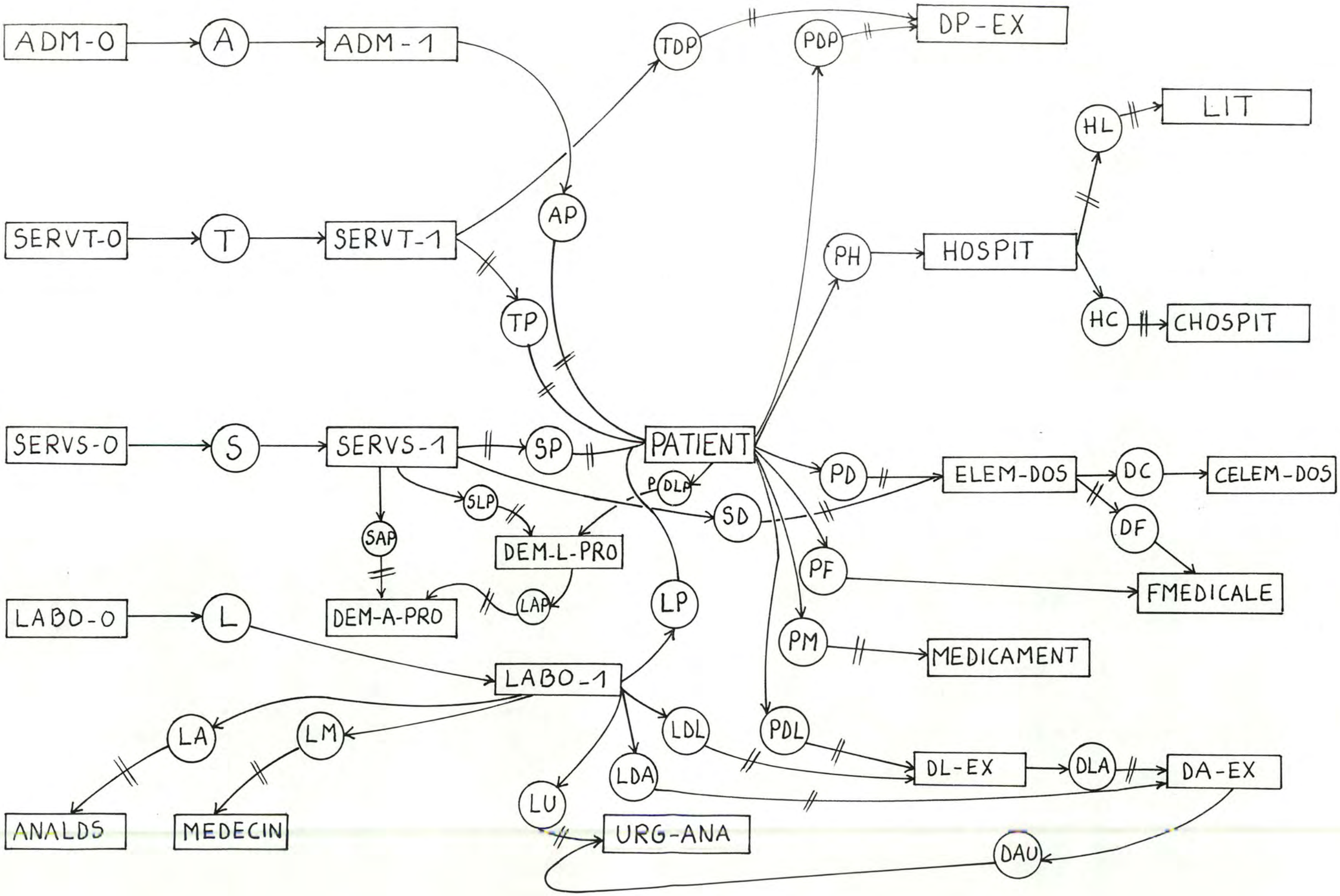
FONCTION FPROTOCOLE



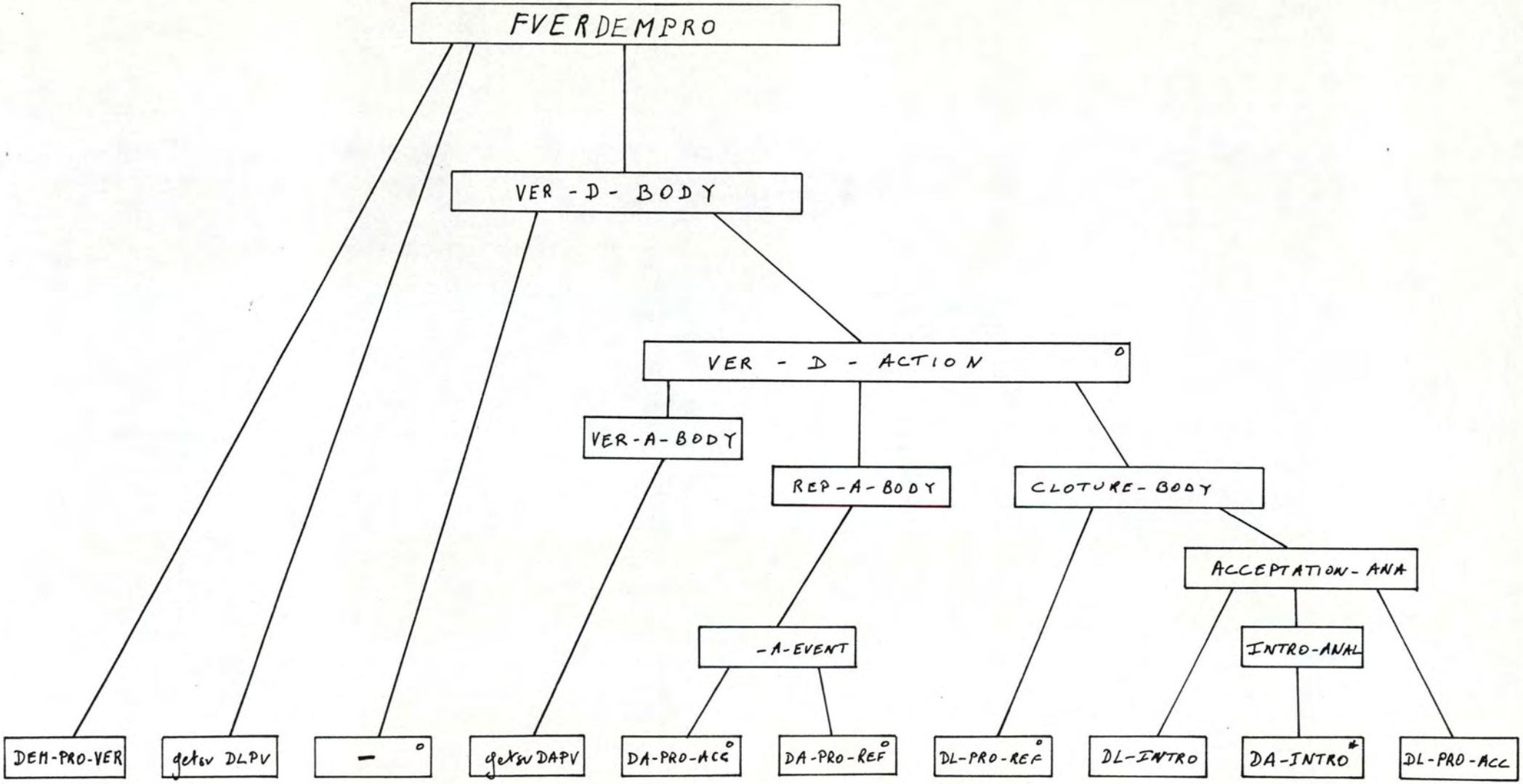
FONCTION FJOBCLIST



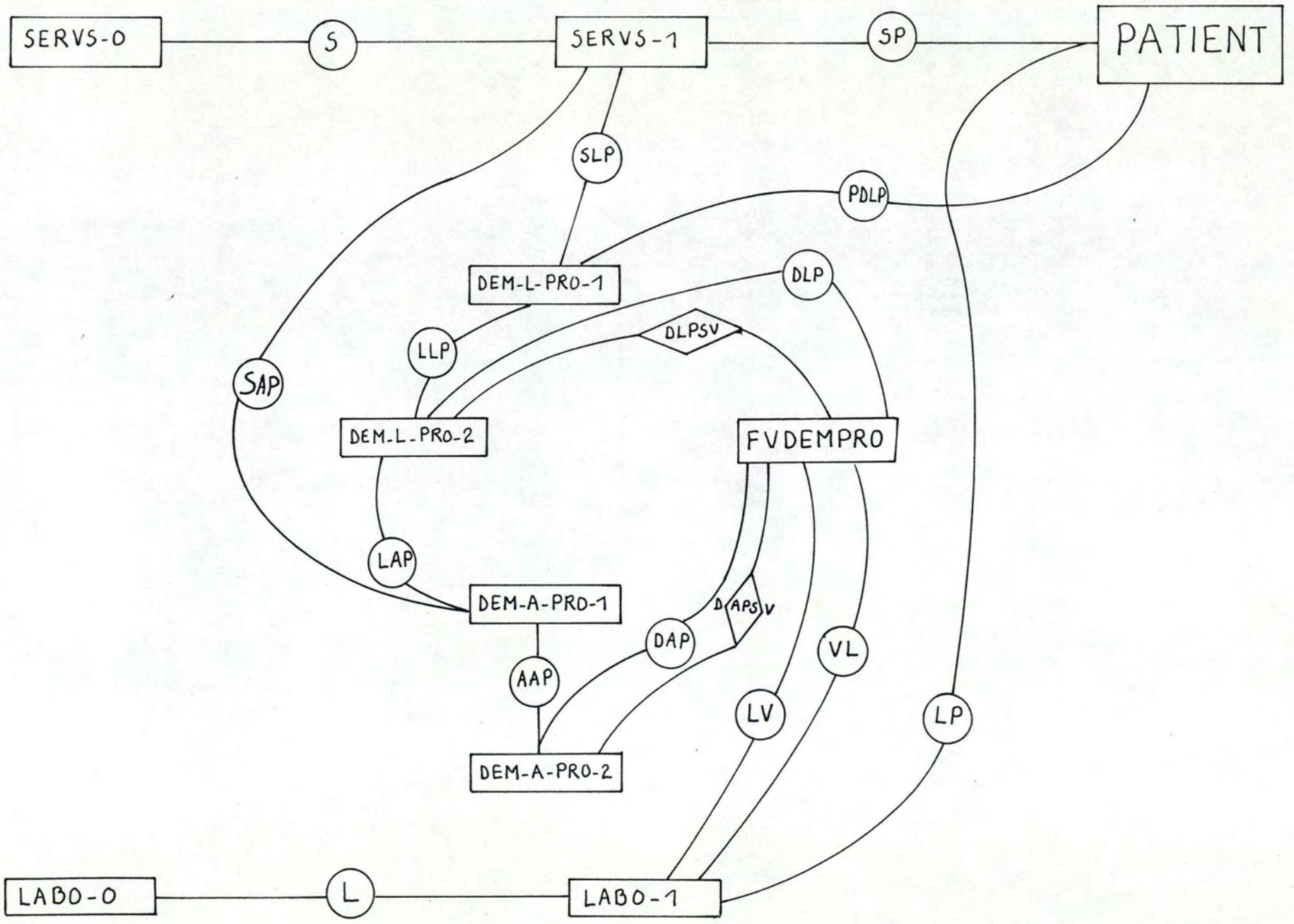
FONCTIONS FHISADM et FHISMED


ADM-0
A
ADM-1
SERVT-0
T
SERVT-1
AP
TDP
PDP
DP-EX
HL
LIT
PH
HOSPIT
HC
CHOSPIT
TP
SERVS-0
S
SERVS-1
SP
PATIENT
PD
ELEM-DOS
DC
CELEM-DOS
DF
P
DLP
SD
SLP
SAP
DEM-L-PRO
PF
FMEDICALE
LABO-0
L
DEM-A-PRO
LAP
LP
PM
MEDICAMENT
LABO-1
LA
LM
LDL
PDL
DL-EX
DLA
DA-EX
LDA
LU
ANALDS
MEDECIN
URG-ANA
DAU

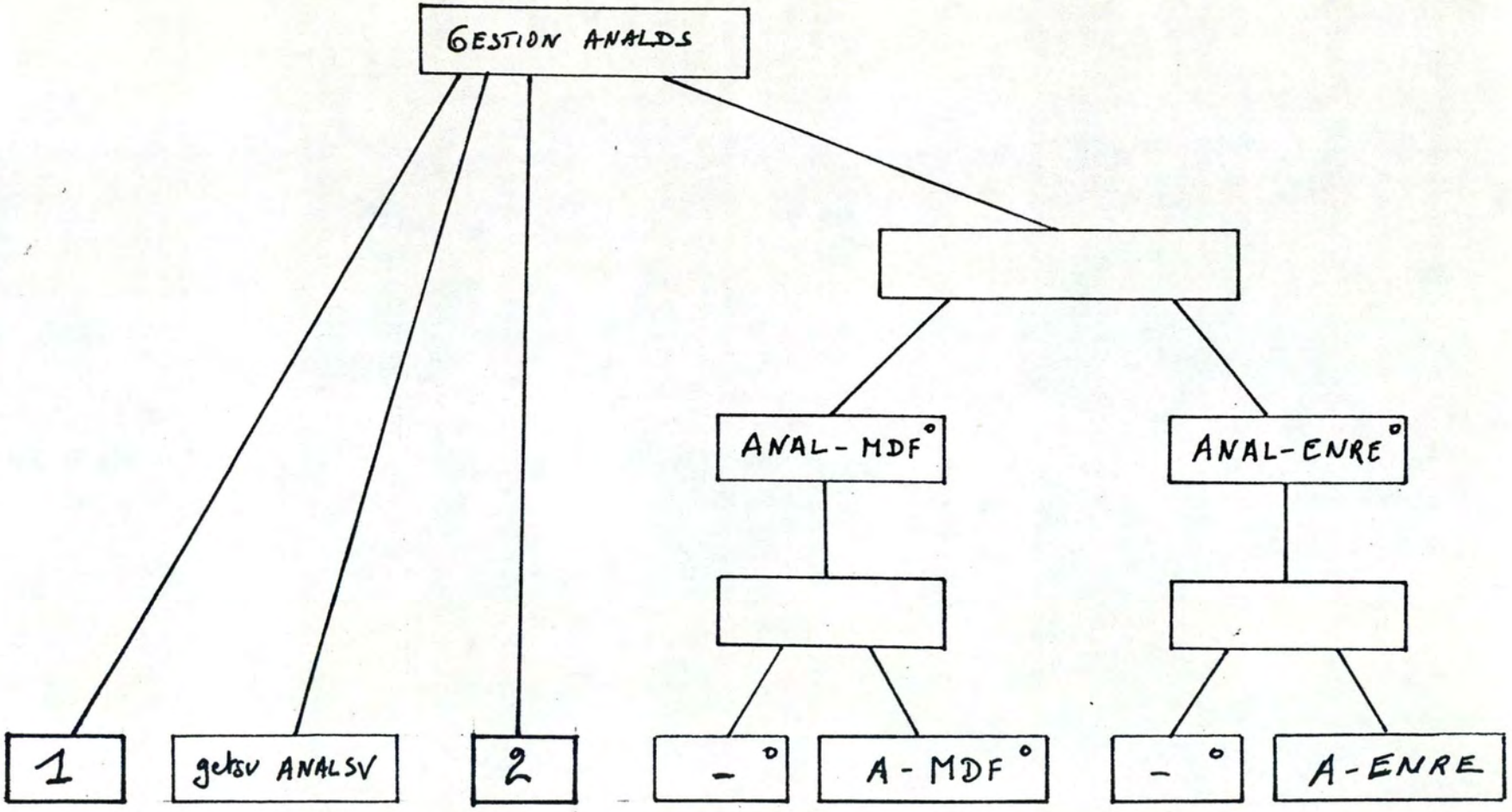
SSD I



FONCTION : Vérifications des demandes provenant

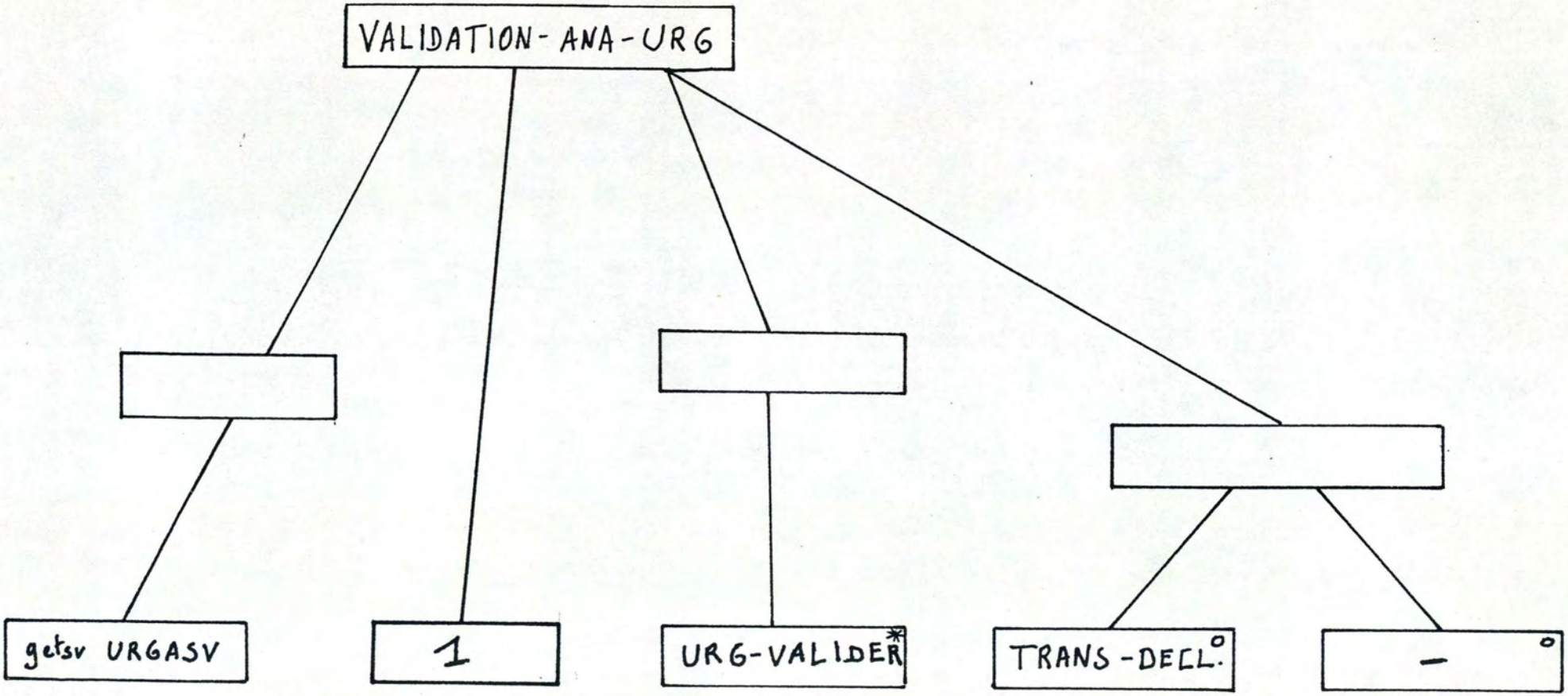

SERVS-0
S
SERVS-1
SP
PATIENT
SLP
PDLP
DEM-L-PRO-1
DLP
LLP
DLPSV
SAP
DEM-L-PRO-2
FVDEMPRO
LAP
DEM-A-PRO-1
DAPSV
DAP
VL
AAP
LV
LP
DEM-A-PRO-2
LABO-0
L
LABO-1

(II.3.4.4.1)

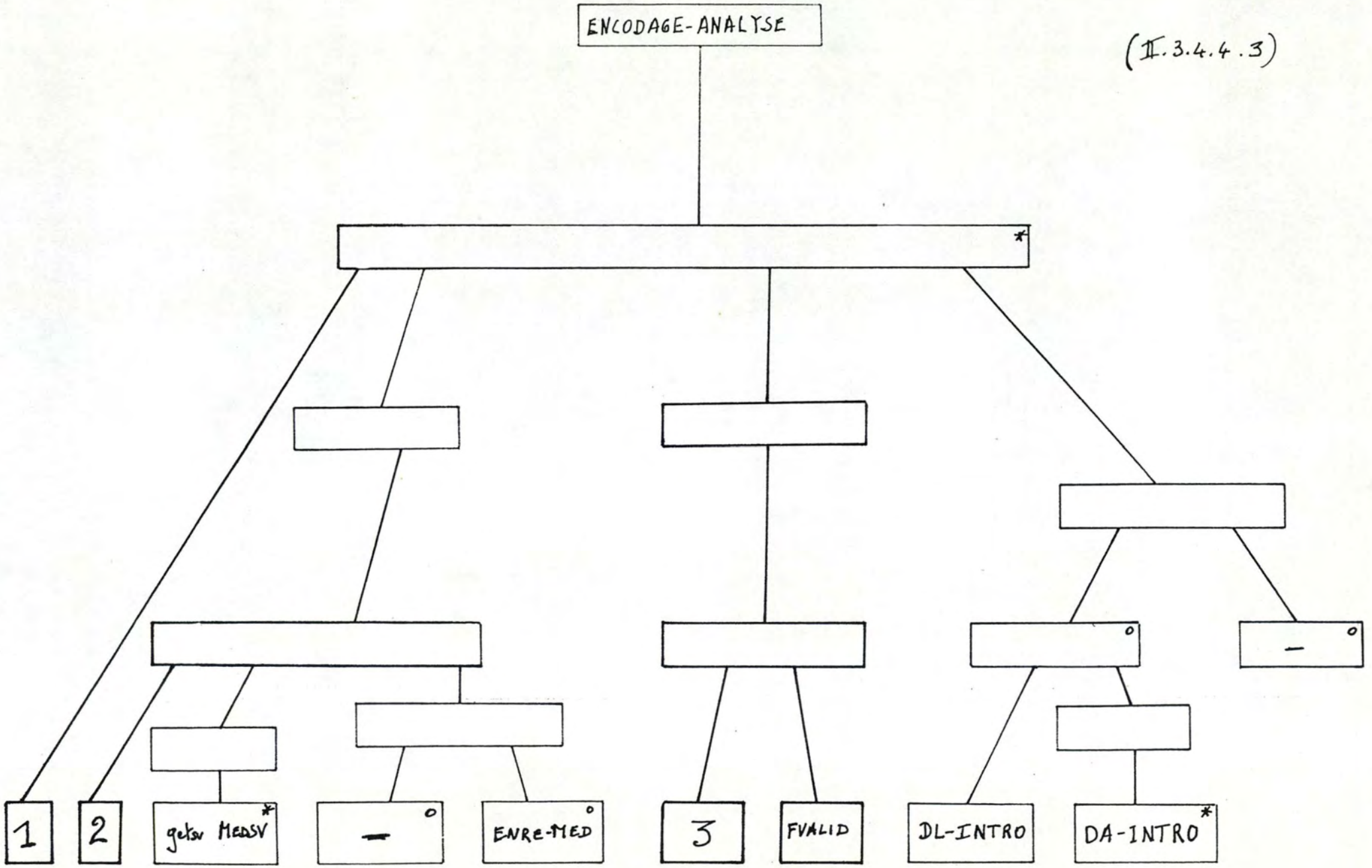


INTERFACE MODÈLE / MONDE REEL

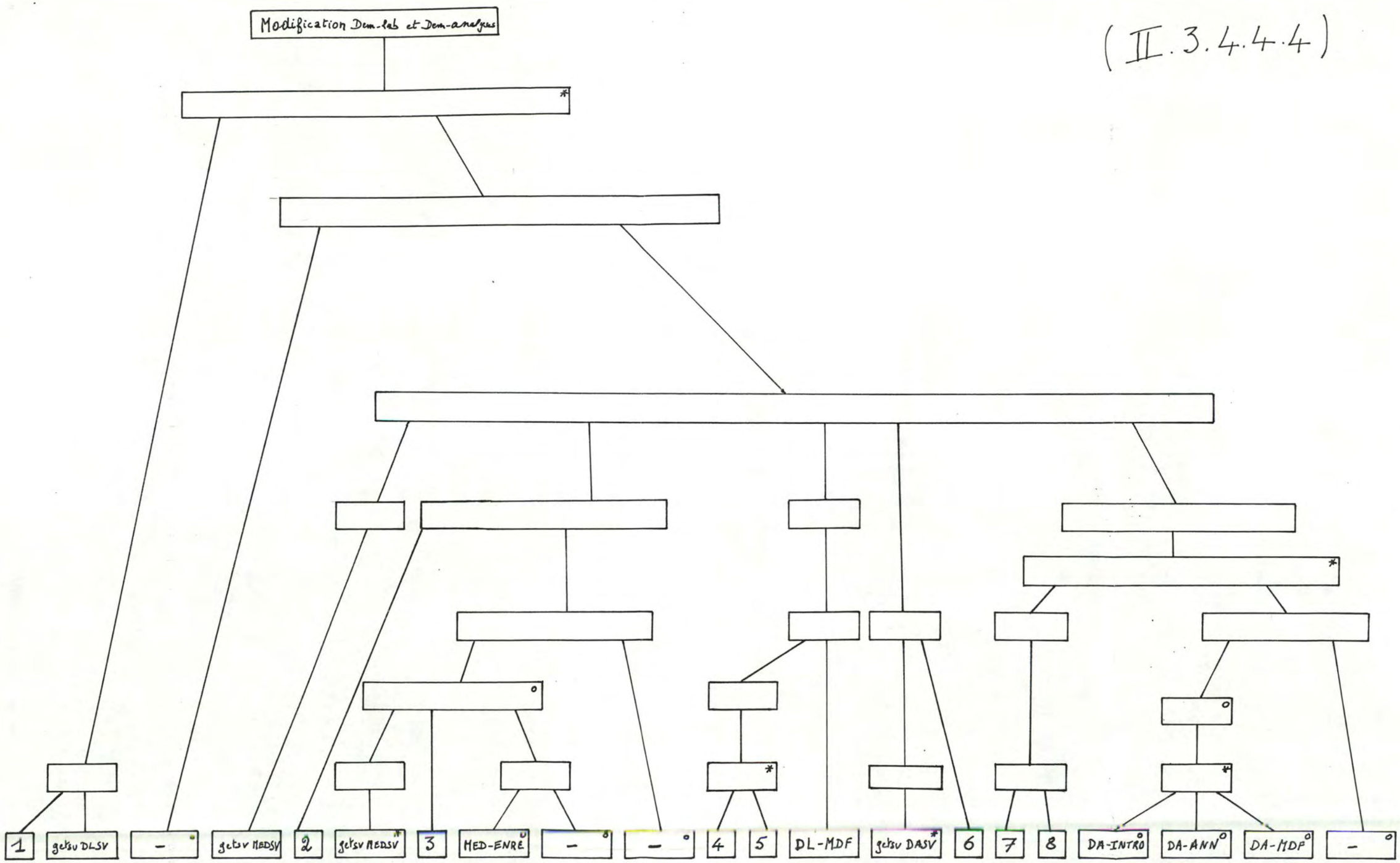
(II. 3.4.4.2)



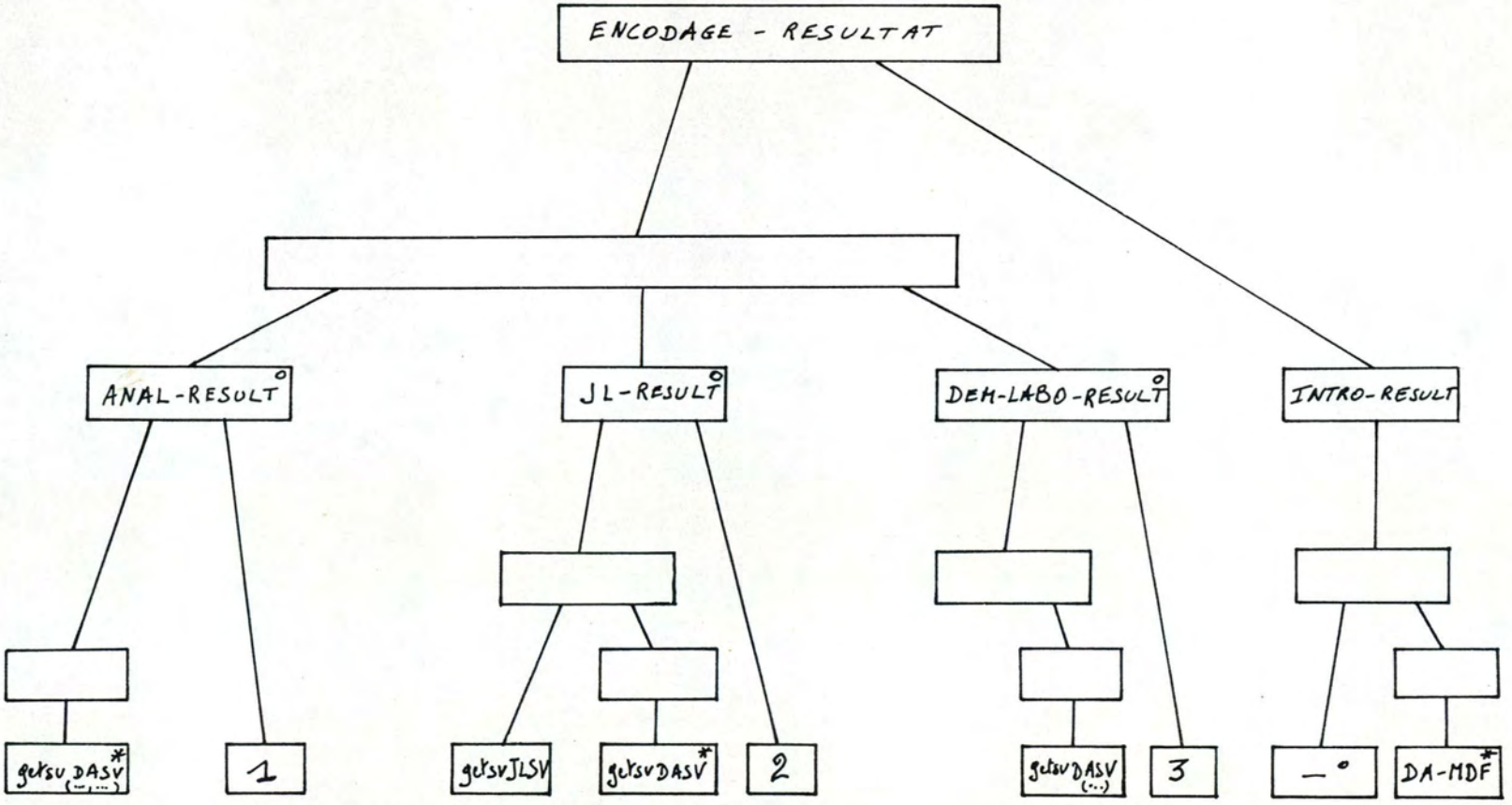
(II.3.4.4.3)

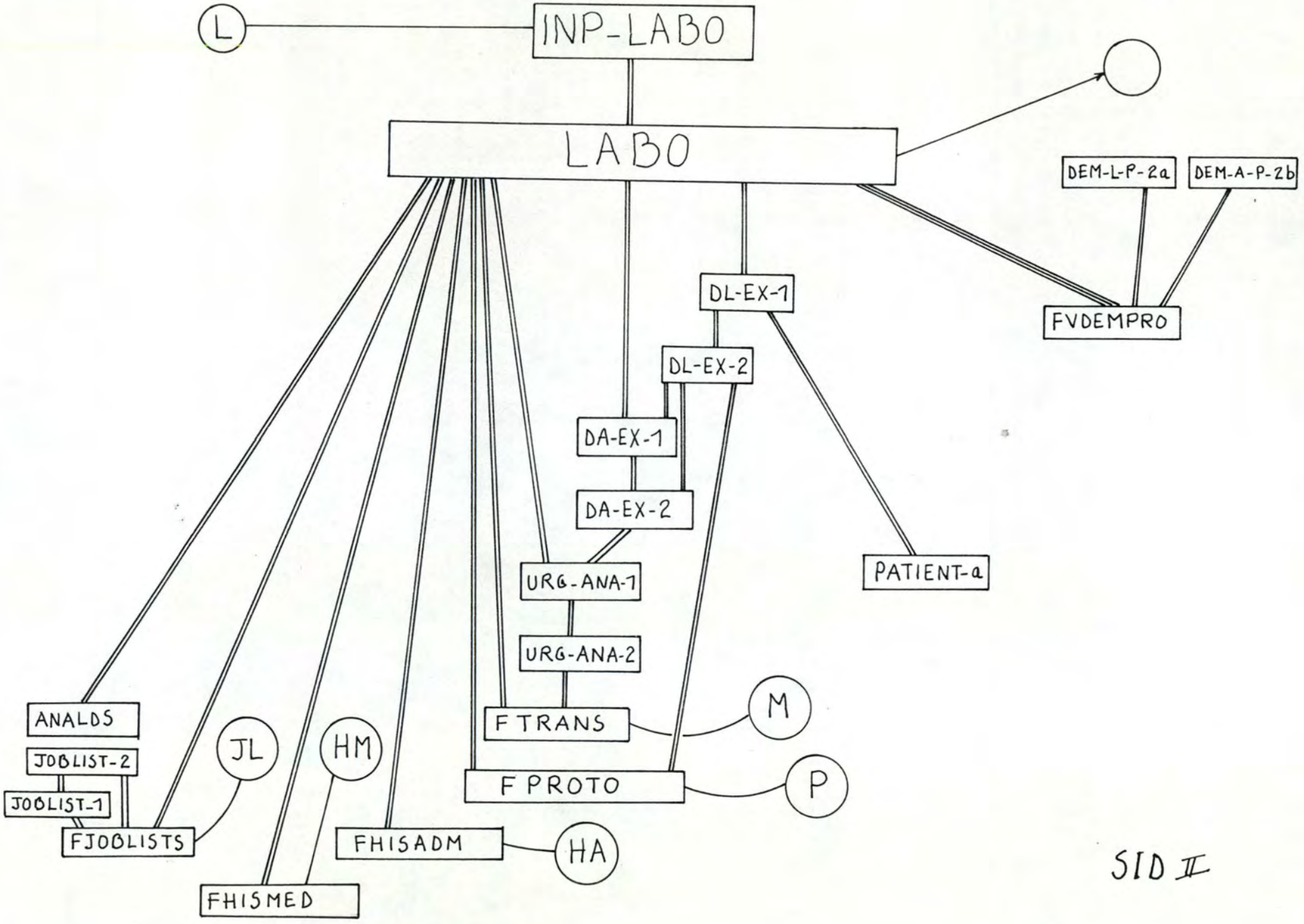

ENCODAGE-ANALYSE
*
1
2
getsv MEDSV
*
—
°
ENRE-MED
°
3
FVALID
°
—
°
DL-INTRO
DA-INTRO
*

(II.3.4.4.4)

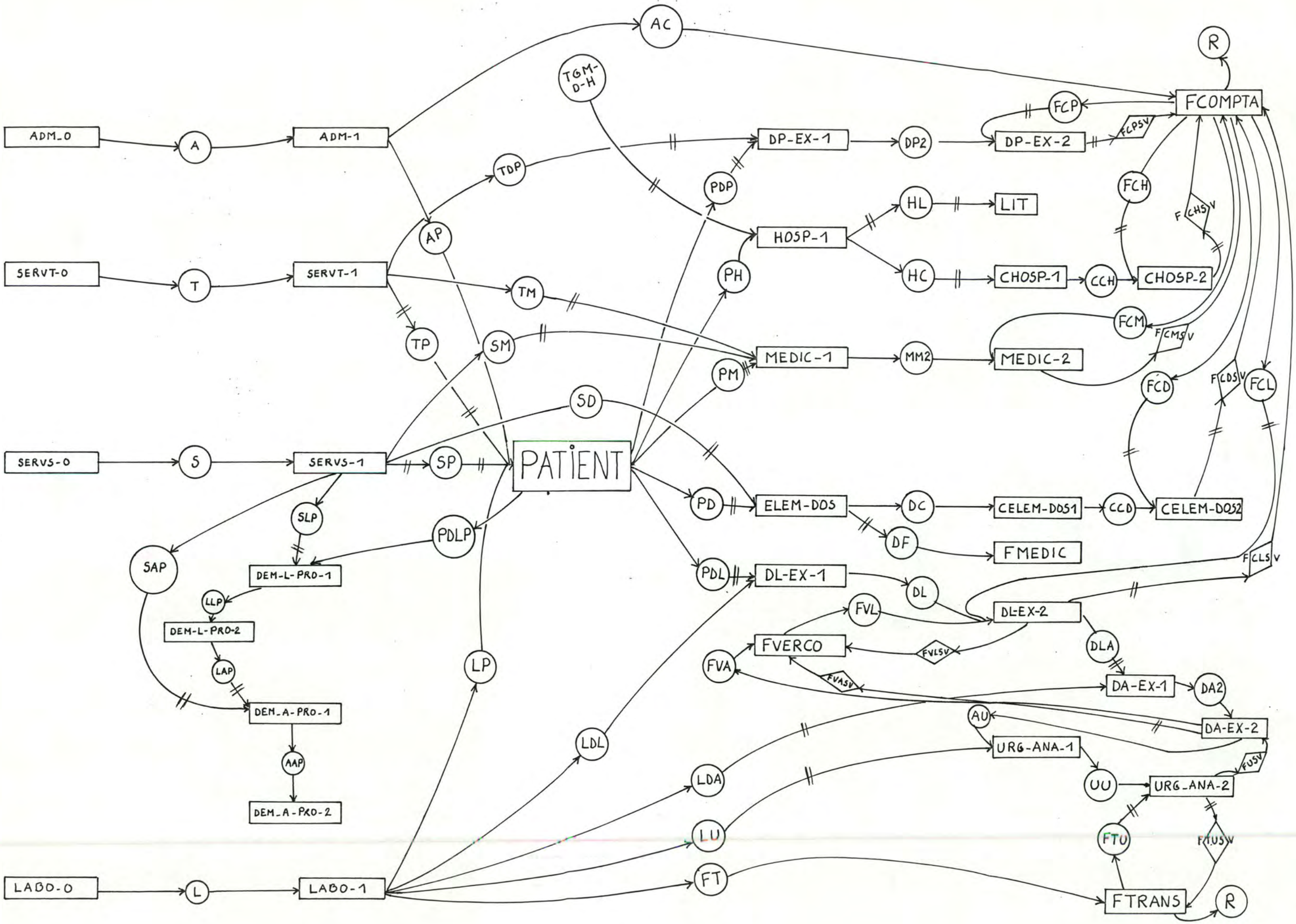

Modification Dem-lab et Dem-analyses
1
getsv DLSV
-
getsv MEDSV
2
getsv MEDSV
3
MED-ENRE
-
-
4
5
DL-MDF
getsv DASV
6
7
8
DA-INTRO
DA-ANN
DA-MDF
-

(II.3.4.4.5)

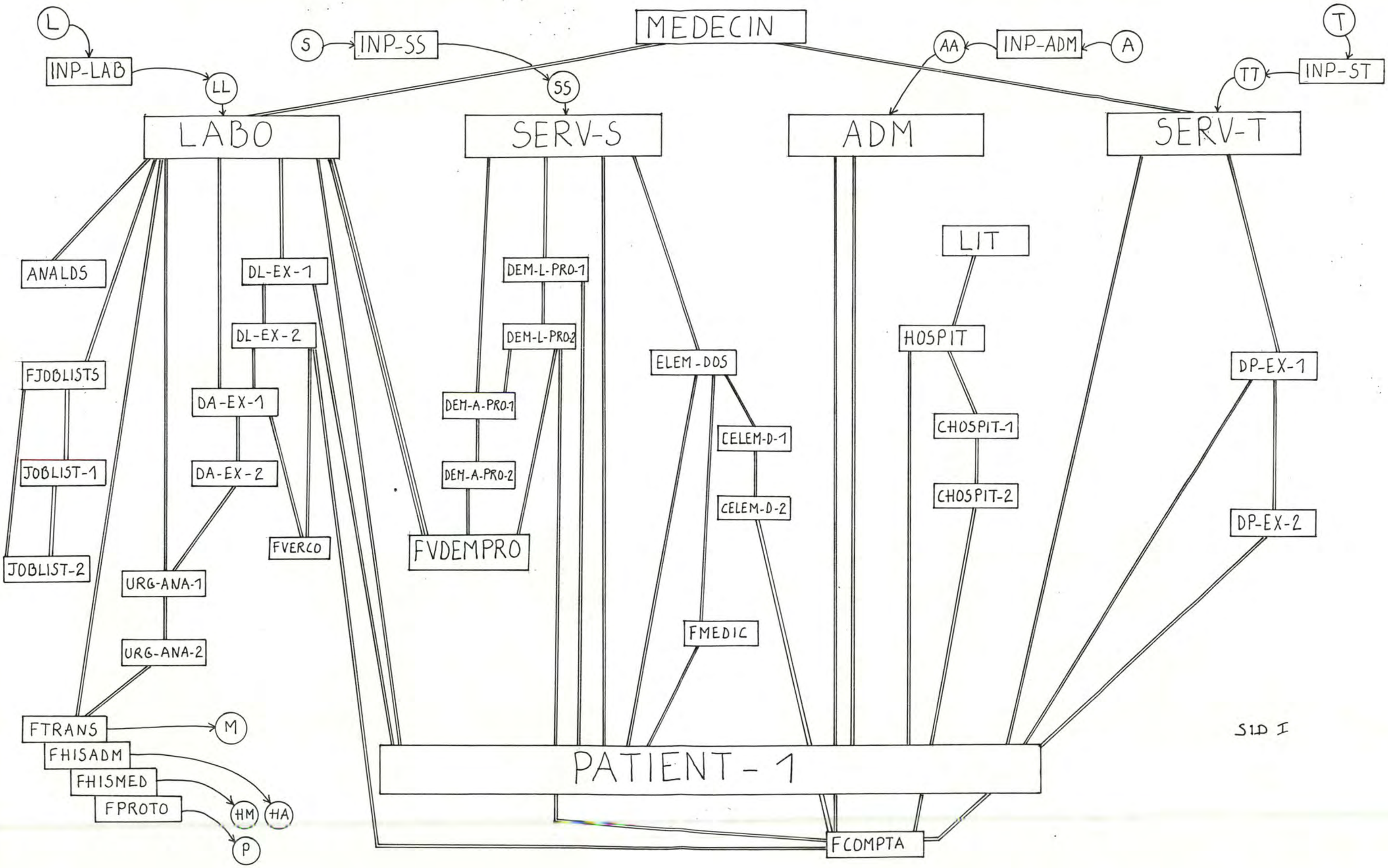

ENCODAGE - RESULTAT
ANAL-RESULT°
JL-RESULT°
DEM-LABO-RESULT°
INTRO-RESULT
1
2
3
—°
DA-MDF*


L
INP-LABO
LABO
DEM-L-P-2a
DEM-A-P-2b
FVDEMPRO
DL-EX-1
DL-EX-2
DA-EX-1
DA-EX-2
URG-ANA-1
URG-ANA-2
PATIENT-a
ANALDS
JOBLIST-2
JOBLIST-1
FJOBLISTS
JL
HM
F TRANS
M
F PROTO
P
FHISADM
HA
FHISMED

SID II


ADM_0
A
ADM-1
AC
TGM-D-H
R
FCOMPTA
FCP
FCPSV
DP-EX-1
DP2
DP-EX-2
TDP
PDP
FCH
FCHSV
HL
LIT
HOSP-1
AP
SERVT-0
T
SERVT-1
TM
PH
HC
CHOSP-1
CCH
CHOSP-2
TP
SM
FCM
FCMSV
MEDIC-1
MM2
MEDIC-2
PM
SD
FCD
FCDSV
FCL
SERVS-0
S
SERVS-1
SP
PATIENT
SLP
PDLP
PD
ELEM-DOS
DC
CELEM-DOS1
CCD
CELEM-DOS2
DF
FMEDIC
SAP
DEM-L-PRO-1
PDL
DL-EX-1
DL
FCLSV
LLP
DEM-L-PRO-2
FVL
DL-EX-2
FVERCO
FVLSV
FVA
FVASV
DLA
LAP
DEM_A-PRO-1
DA-EX-1
DA2
DA-EX-2
AU
LP
LDL
URG-ANA-1
LDA
UU
URG_ANA-2
FUSV
LU
FTU
FTUSV
LABO-0
L
LABO-1
FT
FTRANS
R
DEM_A-PRO-2
AAP

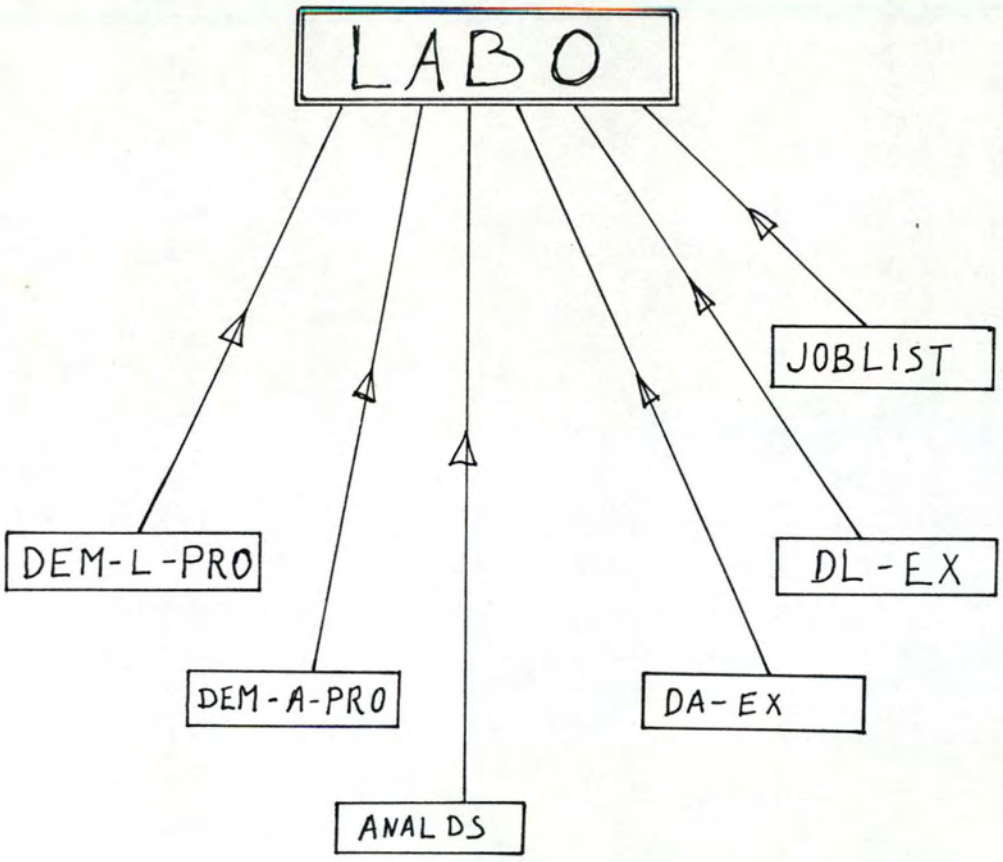
SSD II

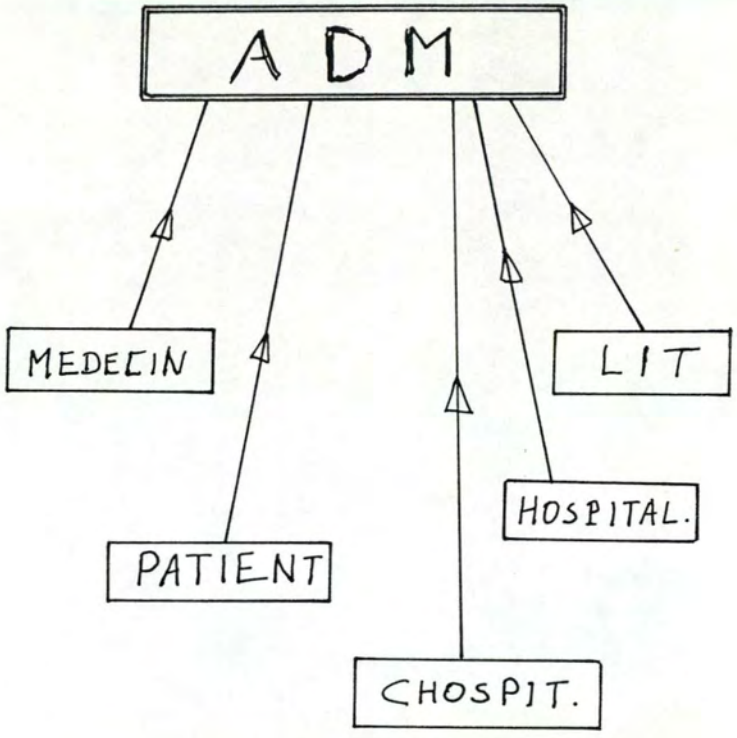

L
INP-LAB
LL
S
INP-SS
SS
MEDECIN
AA
INP-ADM
A
T
TT
INP-ST
LABO
SERV-S
ADM
SERV-T
ANALDS
FJOBLISTS
JOBLIST-1
JOBLIST-2
DL-EX-1
DL-EX-2
DA-EX-1
DA-EX-2
FVERCO
URG-ANA-1
URG-ANA-2
FTRANS
FHISADM
FHISMED
FPROTO
M
HM
HA
P
DEM-L-PRO-1
DEM-L-PRO-2
DEM-A-PRO-1
DEM-A-PRO-2
FVDEMPRO
ELEM-DOS
CELEM-D-1
CELEM-D-2
FMEDIC
LIT
HOSPIT
CHOSPIT-1
CHOSPIT-2
DP-EX-1
DP-EX-2
PATIENT-1
FCOMPTA

SID I

L
INP-LAB
LABO
PAT-a
DL-EX-1
DL-EX-2a
DA-EX-1
DA-EX-2
FVERCO
URG-AN-1
URG-AN-2
UASV
FTRANS
M
DLXSV
DAXSV
FHISADM
HA
FPROTO
P
FHISMED
HM
ANALDS
ADSSV
JLSV
FJOBLIST
JOBLIST-1
JOBLIST-2
D PSV
DAPSV
FDEMP
DEM-A-P-2b
DEM-L-P-2b
INPUT-MED
MEDECIN
MEDSV
A
INP-ADM
ADM
PATSV
PATIENT
HOSP
LIT
LITSV
HOSPSV
CHOSPSV
CHOSP-1
CHOSP-2
FCOMPTA
FT
DL-EX-2b
S
INP-SS
SERV-S
DEM-L-P-1
DEM-L-P-2a
DEM-A-P-1
DEM-A-P-2a

SID III


LABO
JOBLIST
DEM-L-PRO
DL-EX
DEM-A-PRO
DA-EX
ANAL DS


ADM
MEDECIN
LIT
HOSPITAL.
PATIENT
CHOSPIT.